

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ТЕПЛОЕНЕРГЕТИЧНИЙ ФАКУЛЬТЕТ
Кафедра автоматизації проектування енергетичних процесів і систем

«На правах рукопису»
УДК _____

«До захисту допущено»

Завідувач кафедри

(підпис) О.В. Коваль
(ініціали, прізвище)
“ ____ ” _____ 2019 р.

Магістерська дисертація

зі спеціальності 121 Інженерія програмного забезпечення
за спеціалізацією Інженерія програмного забезпечення розподілених систем
на тему: Виокремлення об'єктів на відеокадрах

Виконала: студентка 6 курсу, групи ТВЗ-81мп
(шифр групи)

Круглик Діана Сергіївна
(прізвище, ім'я, по батькові) _____ (підпис)

Науковий керівник к.т.н., доцент Шаповалова С.І.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) _____ (підпис)

Консультант _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) _____ (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) _____ (підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студентка _____
(підпис)

Київ – 2019 року

**Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 121 Інженерія програмного забезпечення

за спеціалізацією – Інженерія програмного забезпечення розподілених систем

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис) О.В. Коваль

” ” _____ 2019р.

ЗАВДАННЯ
на магістерську дисертацію студентці
Круглик Діані Сергіївні

(прізвище, ім'я, по батькові)

1. Тема роботи Виокремлення об'єктів на відеокадрах

керівник роботи Шаповалова Світлана Ігорівна, к.т.н., доцент

(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ” ” 201р. №

2. Строк подання студентом роботи грудня 2019р.

3. Об'єкт дослідження

4. Предмет дослідження

5. Перелік питань, які потрібно розробити провести аналіз існуючих методів і досліджень, що були проведені по вибраній темі виокремлення об'єктів на відеокадрах; запропонувати схему етапів вирішення задачі вищначення об'єктів на зображенні за допомогою бібліотеки OpenCV та фреймворку Tensorflow; проаналізувати методи виокремлення об'єктів на відеокадрах такі як нейронні мережі та каскадні класифікатори; перевірити роботу обох методів на тестових даних і порівняти результати для визначення кращого з них.

6. Орієнтовний перелік ілюстративного матеріалу: приклад роботи програмних забезпечень виокремлення об'єктів на відеокадрах; огляд роботи методу прибирання шумів із зображення; ілюстрація сегментації зображення; наведення прикладів ознак Хаара; наведена схема тренування каскадів; модулі бібліотеки OpenCV; ілюстрації різних типів нейронних мереж; схема розв'язання задачі з використанням бібліотеки OpenCV; схема розв'язання задачі з використанням фреймворку Tensorflow; приклади роботи бібліотеки OpenCV на тваринах, людях,

птахх; схема етапів рішення задачі з використанням Tensorflow; результати роботи програмного забезпечення тестового завдань з використанням фреймворку Tensorflow та бібліотеки OpenCV.

7. Орієнтований перелік публікацій Виокремлення об'єктів на відеокадрах (“Сучасні проблеми наукового забезпечення енергетики” XVII міжнародної науково-практичної конференції аспірантів, магістрів, студентів)

8. Дата видачі завдання ” ____ ” _____ 2019 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	28.09.18 р.	
2	Опрацювання літературних джерел	01.10.18 р. – 03.02.19 р.	
3	Підготовка матеріалів дисертації	04.02 – 31.05.19 р.	
4	Підготовка доповідей на конференції	11.03 – 29.03.19 р.	
6	Розробка програмного продукту	03.06 – 25.10.19 р.	
5	Переддипломна практика	02.09 – 25.10.19 р.	
7	Захист програмного продукту	26.10.19 р.	
8	Розробка стартап-проекту	11.11 – 19.11.19 р.	
9	Передзахист	20.11.19 р.	
10	Оформлення дисертації	21.11- 29.11.19 р.	
11	Захист		

Студентка

_____ (підпис)

Круглик Д.С.

_____ (прізвище та ініціали,)

Керівник роботи

_____ (прізвище та ініціали,)

Шаповалова С.І.

_____ (підпис)

РЕФЕРАТ

Структура і обсяг дипломної роботи

Магістерська дисертація складається зі вступу, п'яти розділів і висновку. Робота містить в собі 18 джерела за переліком посилань, 26 ілюстрацій, 24 таблиць. Крім того, в кінці розміщено 1 додаток. Основна частина роботи викладена на 91 сторінках.

Актуальність теми

Комп'ютерний зір існує вже більше 50 років, але останнім часом ми спостерігаємо велике виникнення інтересу до того, як машини "бачать" і як комп'ютерне бачення можна використовувати для створення продуктів для споживачів та бізнесу. Кілька прикладів таких додатків — Amazon Go, Google Lens, автономні транспортні засоби, розпізнавання обличчя.

Ключовим рушійним фактором, що стоїть у всьому цьому, є комп'ютерний зір. Простіше кажучи, комп'ютерний зір — це дисципліна в рамках широкої області штучного інтелекту, яка вчить машини бачити. Його мета - витягнути значення з пікселів.

З біологічної точки зору, його метою є створення обчислювальних моделей зорової системи людини. З інженерної точки зору, комп'ютерний зір має на меті побудувати автономні системи, які могли б виконувати деякі такі завдання, які може виконувати візуальна система людини (і навіть перевершувати її у багатьох випадках).

Дана робота присвячена розробці схеми етапів розв'язання задачі виокремлення об'єктів на відекадрах.

Метою дослідження є створення схеми етапів розв'язання задачі виокремлення об'єктів на відеопотоках, на прикладі тестових задач визначення та підрахунок машин на дорогах різними способами.

Для досягнення поставленої мети потрібно виконати наступні **задачі дослідження:**

— провести аналіз існуючих методів і досліджень, що були проведені по вибраній темі виокремлення об'єктів на відеокадрах;

— на основі проведеного раніше аналізу запропонувати схему етапів вирішення задачі вищначення об'єктів на зображенні за допомогою бібліотеки OpenCV та фреймворку Tensorflow;

— розглянути програмні засоби, що можуть вирішувати задачу виокремлення об'єктів на відеокадрах;

— проаналізувати методи виокремлення об'єктів на відеокадрах такі як нейронні мережі та каскадні класифікатори;

— перевірити роботу обох методів на тестових даних і порівняти результати для визначення кращого з них;

Об'єкт дослідження: узагальнений підхід до розв'язування задач виокремлення об'єктів на відеокадрах з використанням бібліотеки OpenCV та фреймворку Tensorflow.

Предмет дослідження: розпізнавання об'єктів за допомогою використання нейронних мереж та каскадних методів.

Методи дослідження: каскадні методи класифікації та нейронні мережі.

Наукова новизна: було розроблено схему етапів розв'язання задачі виокремлення об'єктів на відеопотоках різними методами.

Практичне значення: запропонований зрозумілу схему етапів розв'язання задачі визначення об'єктів на відеокадрах, що дає змогу розробникам значно скоротити та спростити час розробки власних програмних забезпечень.

Апробація: результати досліджень представлені на XVII міжнародній науково-практичній конференції аспірантів, магістрів, студентів на тему «Сучасні проблеми наукового забезпечення енергетики» 2019 року.

Ключові слова: СЕГМЕНТАЦІЯ, КАСКАДНІ КЛАСИФІКАТОРИ, ВИОКРЕМЛЕННЯ ОБ'ЄКТІВ, РЕГІОНАЛЬНО-ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, МЕТОДИ СЕГМЕНТАЦІЇ.

ABSTRACT

Structure and volume of the thesis

The master's thesis is best gathered, five divided and offered. The work is contained in 18 sources for viewing, 26 pictures, 24 tables. In addition, there is 1 app at the end. The main work is on 91 pages.

Actuality of theme

Computer vision has been around for over 50 years, but we offer more than we have done a lot of growing and then "machines" see "as a computer if it requires use for consumers and users. A few additional extras — Amazon Go, Google Lens, standalone vehicles, communications extensions.

Computer vision is a key driver in the whole business. Computer stars are a discipline in the broad field of artificial intelligence that machines must contain. Interesting goal — select pixel values.

From a biological point of view, it requires the creation of computational models of the human visual system. From an engineering point of view, a working computer is designed to use an autonomous system that can be used to do what you can do if you want to make sure its person has it (check it with).

This work has devoted to the developers of the scheme of stages of expansion of tasks of separation of objects on video frames.

The research methodology is to create schemes of stages of expansion of tasks of separation of objects on video streams, at application of test tasks which are under action and calculation of machines on expensive different possibilities.

To achieve this goal, you must perform **the following tasks:**

— analyze the analysis of different methods and people who were vibrating the subject of object separation on video frames;

— in support of the leading analysis of the analogues, we propose a scheme of equal results that determine the higher utilization of objects in the picture for the OpenCV library and the Tensorflow framework;

— explore music characteristics using previous working and quality classrooms;

— review the software tools that allow you to perform the tasks of isolating objects in video frames;

— to analyze methods of separation of objects on video frames as neural networks and cascade classifiers;

— test the performance of both methods on test data and compare the data for the best ones;

About Scientific Research: A Generalized Approach to Expanding Object Recovery Objects on Video Frames from the OpenCV Library and the Tensorflow Framework.

Study Exploration: Object Discovery for the Use of Neutron Measures and Cascade Methods.

Research methodology: cascading classification methods and neural networks.

Scientific news: A scheme of equal disparate tasks for object separation on video streams of different methods was developed.

Practical value: It is suggested to understand the scheme of the steps of the expansion of the tasks contained in the video frames, which were able to the developers to quickly inform and repeat the time of developing their own software programs.

Testing: Representatives presented at the 18th international scientific and practical conference of postgraduate, masters, students on the topic "Modern problems of the scientific space of energy" in 2019.

Keywords: SEGMENTATION, CASCADE CLASSIFIERS, OBJECT STRENGTH, REGIONAL-CONVERTING NEURAL NETWORK

ЗМІСТ

Перелік умовних позначень, скорочень і термінів.....	10
Вступ.....	11
1 Задача виокремлення об'єктів на відеопотоках	12
1.1 Постановка задачі та місця її застосування	12
1.2 Етапи розв'язання задачі.....	13
1.3 Підготовка зображення	14
1.3.1 Очищення від шуму	15
1.3.2 Деблюризація.....	17
1.3.3 Реєстрація зображення	19
1.4 Виокремлення об'єктів	21
1.4.1 Сегментація.....	21
1.4.2 Мумфорд-шахський підхід	23
1.4.3 Геодезичні контури	24
1.4.4 Методи сегментації	26
1.4.5 Алгоритм R-CNN.....	29
Висновки до розділу 1	30
2 Програмне забезпечення розв'язання задачі виокремлення об'єктів	31
2.1 Бібліотека OpenCV	31
2.2 Фреймворк Tensorflow	36
Висновки до розділу 2	48
3 Схема розв'язання задачі	50
3.1 Схема розв'язання на основі OpenCV	50
3.2 Схема розв'язання на основі Tensorflow	54
Висновки до розділу 3	56
4 Обчислювальні експерименти	57
4.1 Тестова задача	57

4.2 Результат роботи та порівняння застосованих підходів	57
Висновки до розділу 4	60
5 Розробка стартап проєкту	61
5.1 Опис ідеї проєкту	61
5.2 Технологічний аудит ідеї проєкту	64
5.3 Аналіз ринкових можливостей запуску стартап-проєкту	66
5.4 Розробка ринкової стратегії проєкту	74
5.5 Розроблення маркетингової програми	78
Висновки до розділу 5	82
Висновки.....	84
Список використаних джерел.....	85
Додаток А	87

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

BV	–	bounded variation
TV	–	total variation
PDE	–	partial differential equation
AOS	–	additive operator splitting
GCS	–	globally convex segmentation
SSD	–	sum of squared differences
ROF	–	Rudin, Osher, and Fatemi
CDF	–	Chambolle's dual formulation
DSS	–	distance selective segmentation
CDSS	–	convex distance selective segmentation
VMS	–	variant Mumford-Shah
SBF	–	stabilised bias field
FSP	–	fitting shape prior DSP distance shape prior
PCV	–	Poisson deblurring followed by CV segmentation
BSK	–	Bar, Sochen, and Kiryati

ВСТУП

Розпізнавання об'єктів - це загальний термін для опису сукупності пов'язаних із цим завдань комп'ютерного зору, які передбачають ідентифікацію об'єктів на цифрових фотографіях.

Класифікація зображень передбачає прогнозування класу одного об'єкта в зображенні. Локалізація об'єкта означає визначення місця розташування одного або декількох об'єктів у зображенні та нанесення рядового поля навколо їх розміру. Виявлення об'єктів поєднує ці дві задачі та локалізує та класифікує один або кілька об'єктів у зображенні.

Ще одним розширенням завдань комп'ютерного зору є сегментація об'єктів, яка також називається "сегментація екземпляра об'єкта" або "семантична сегментація", де екземпляри розпізнаних об'єктів позначаються виділенням конкретних пікселів об'єкта замість грубої рамки обмеження.

На теперішній час існує багато методів та готових бібліотек чи фреймворків, які розроблені для того, що виокремлювати об'єкти на відеокадрах або статичних зображеннях.

Однак, на жаль деякі з них не пропонують достатньо високу точність виконання програми, або ж коштують надто дорого для звичайного користувача, студента чи навіть компанії. Тож необхідно розглянути та визначити декілька методів, завдяки яким можна побудувати власний додаток, що зможе розпізнавати будь-який об'єкт по бажанню користувача.

Для цього потрібно побудувати схему етапів розв'язання подібних задач, щоб пришвидшити та спростити роботу розробника, та запропонувати йому недорогий та якісний метод виокремлення об'єктів на зображенні.

1 ЗАДАЧА ВИОКРЕМЛЕННЯ ОБ'ЄКТІВ НА ВІДЕОПОТОКАХ

Метою даної дипломної роботи є розв'язання задачі виокремлення об'єктів на відеокадрах, а саме розроблення рішення у вигляді схеми використання програмного забезпечення всіх етапів розв'язання задач із визначенням відповідного програмного забезпечення.

Буває, кожен із нас витратив на пошуки загублених ключів від кімнати купу часу вдома? Це трапляється з кожним, і на сьогоднішній день це все ще неймовірно розчаровуючий досвід. Але що, якщо простий комп'ютерний алгоритм міг би знайти ваші ключі за лічені мілісекунди?

Це сила алгоритмів виявлення об'єктів. Хоча це був простий приклад, програми виявлення об'єктів охоплюють різноманітні та різноманітні галузі: від цілодобового спостереження до виявлення транспортних засобів у режимі реального часу у містах. Коротше кажучи, це потужні алгоритми глибокого навчання.

У цій дипломній роботі ми заглибимося глибше і розглянемо різні алгоритми, які можна використовувати для виявлення об'єктів. Це можуть бути алгоритми, що належать до сімейства RCNN, тобто RCNN, Fast RCNN і Fastter RCNN. Також існують більш вдосконалені алгоритми, такі як YOLO, SSD тощо, окрім цього бібліотека OpenCV, фреймворк Tensorflow [12].

1.1 Постановка задачі та місця її застосування

Така задача є дуже затребуваною у сучасному світі та знаходить використання у багатьох галузях. Необхідність визначення об'єкту на відеопотоках в прикладних задачах може бути у визначенні небажаних осіб чи віп клієнтів публічних місцях,

зібрані статистичні дані для подальшого аналізу, наприклад визначення в який день приходить найбільше відвідувачів, як показано на рисунку 1.1



Рисунок 1.1 – Демонстрація роботи визначення осіб на зображенні

Окрім цього, проведення статистики за відеопотоками може бути застосоване у цілях безпеки на складах, у лікарнях, офісах, лабораторіях тощо. Також затребуваною є задача визначення машин на дорогах і їх підрахунок, та на паркінгах для визначення вільних та зайнятих місць. Окрім цього, може бути використаним у заповідниках для відстежування і контролю тварин.

1.2 Етапи розв'язання задачі

Розглянемо етапи розв'язання задачі. У даній дипломній роботі розглядається два способи вирішення задачі виокремлення об'єкта на зображенні: за допомогою каскадного методу на нейронних мережах.

Першим кроком необхідно підготувати зображення, а саме очистити зображення від шумів, згладити його. Для цього можна використати бібліотеку OpenCV. Після цього крок розпізнавання об'єктів. Існує декілька способів це

виконати, за допомогою пошуку контурів, нейронних мереж чи каскадних класифікаторів, як показано на рисунку 1.2.

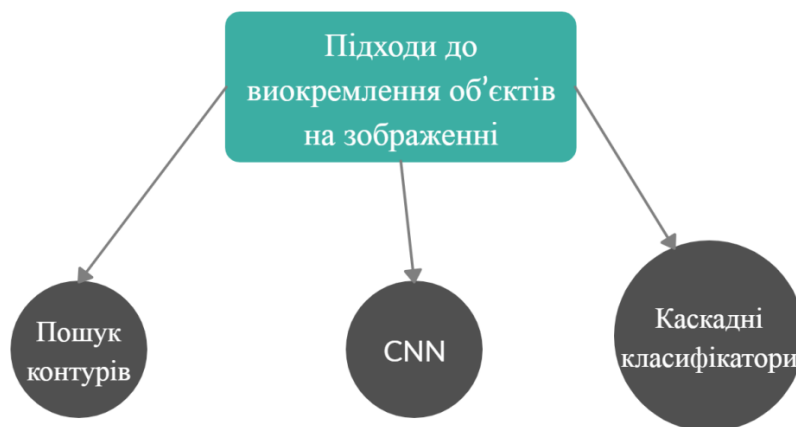


Рисунок 1.2– Підходи виокремлення об'єктів на зображенні

Для програмування програмного продукту із використанням каскадних методів використовувалася бібліотека OpenCV. Вона являє собою дуже поширений інструмент обробки зображень та пропонує ефективне та досить точне розпізнавання об'єктів за допомогою каскадів Хаара [10].

Висвітliamo етапи роботи програмного забезпечення на основі бібліотеці OpenCV. Першим етапом є завантаження відео на сервер. Після цього необхідно підготувати зображення для подальшої обробки. Наступним етапом є розпізнавання об'єкта на зображенні. Потім необхідно обробити накопичені дані та візуалізувати у вигляді таблиць та графіків.

1.3 Підготовка зображення

Для того, щоб методи виокремлення об'єктів на зображенні давали кращі результати, необхідно обробити та очистити зображення від різного виду пошкоджень.

Існує багато типів шумів, такі як гаусовський, «сіль та перець», пуассон, спекл-шум та інші. Наприклад, розглянемо як впливає шум на якість зображення та як виглядає зображення після очищення від шуму, як показано на рисунку 1.3.



Рисунок 1.3— Зашумлене зображення та після очищення методом TV – загальна варіація

Розглянемо різні методи, що можуть бути використані для покращення зображення.

1.3.1 Очищення від шуму

Шум присутній майже в будь-якому зображенні, часто з'являється при зборі або зберіганні даних. Існує багато різних типів шуму, таких як гауссовий, який має добавки і незалежний від інтенсивності сигналу і Пуассона, який залежить від яскравості.

Розуміння теми очищення від шумів має важливе значення для моделювання її видалення. У межах різних фреймворків семінарна робота про ROF була введена у 1992 році [5]. Це важливо для її здатності видаляти шум при цьому зберігаючи краї, на відміну від попередніх підходів, що використовували L^2 регуляризацію градієнта. Це досягається на прикладі регуляризації TV.

Надається наступна формула:

$$\min_u \left\{ \alpha \int_{\Omega} |\nabla u| dx + \frac{1}{2} \int_{\Omega} |u - z|^2 dx \right\} \quad (1.1)$$

де α - позитивний параметр, що управляє рівнем видалення шуму.

Важливо зазначити, що (1.1) добре складена, і таким чином ми можемо гарантувати існування та унікальність мінімізатора.

Рівняння Ейлера-Лагранжа сформульоване наступним чином(1.2):

$$u - z - \alpha \operatorname{div} \left(\frac{Du}{|Du|} \right) = 0. \quad (1.2)$$

Однак необхідно враховувати випадки, коли $Du = 0$. Це особливо важливо при розгляді $z \in L^\infty(\Omega)$, і цілком можливо, що u буде демонструвати ступінчаті властивості. Маючи це на увазі, ми тепер отримаємо рівняння Ейлера-Лагранжа [8].

Спочатку вводимо таку множину(1.3):

$$K = \{-\operatorname{div} \varphi : \varphi \in C_0^\infty(\Omega; \mathbb{R}^n) : |\varphi(x)| \leq 1 \ \forall x \in \Omega \quad (1.3)$$

і наближену K of K in $L^2(\Omega)$, яка визначається як (1.4):

$$K = \{-\operatorname{div} h : h \in L^\infty(\Omega; \mathbb{R}^n) : -\operatorname{div}(h_{\Omega X}) \in L^2(\mathbb{R}^n) : |h(x)|_\infty \leq 1 \quad (1.4)$$

Також необхідно зазначити що $J \in (1.5)$:

$$J(u) = TV(u) = \sup_{p \in K} \int_{\Omega} u(x)p(x)dx, \quad (1.5)$$

Якщо $u \in L^2(\Omega)$, то

$$J(u) = \sup_{p \in K} \int_{\Omega} u(x)p(x)dx. \quad (1.6)$$

Це приводить до того, що:

$$K = \left\{ p \in L^2(\Omega) : \int_{\Omega} p(x)u(x)dx \leq J(u) \ \forall u \in L^2(\Omega) \right\}, \quad (1.7)$$

де K найбільша множина в $L^2(\Omega)$ така, що (1.9), отримується для будь-якої $u \in L^2(\Omega)$. Отже, згідно з (1.7), для $u \in L^2(\Omega)$:

$$\partial J(u) = \left\{ p \in K : \int_{\Omega} p(x)u(x)dx = J(u) \right\}. \quad (1.8)$$

Тепер ми можемо отримати рівняння (1.9), що задовольняє u (1.8), що мінімізує ROF:

$$\alpha J(v) + \frac{1}{2} \int_{\Omega} (v - z)^2 dx \geq \alpha J(u) + \frac{1}{2} \int_{\Omega} (v - z)^2 dx. \quad (1.9)$$

Це, у свою чергу, демонструє (1.10), що:

$$\alpha J(v) \geq \alpha J(v) + \int_{\Omega} (v - u)(z - u)dx - \frac{1}{2} \int_{\Omega} (u - v)^2 dx. \quad (1.10)$$

Далі, для будь-якої $t \in \mathbb{R}$ (1.11):

$$\alpha J(u + t(v - u)) - J(u) - t \int_{\Omega} (v - u)(z - u) dx \geq 0. \quad (1.11)$$

Це показує, що (1.12):

$$\frac{z-u}{\alpha} \in \partial J(u). \quad (1.12)$$

Отримані формули приводять нас до рівняння Ейлера-Лагранжа (1.13) для моделі ROF:

$$u - z + \alpha \partial J(u) \quad (1.13)$$

Незважаючи на її успіх і постійне значення в літературі, результати продемонстрували так званий ефект сходів, де реконструйовані гладкі області як кусково-постійні і, таким чином, обмежують якість.

1.3.2 Деблюризація

Важливо врахувати інший тип погіршення зображення. В той час як процес деноїзації включений в багато сучасних методів сегментації, коли зображення розмиті, воно часто виходить за рамки звичайних методів сегментації. Розмивання зображення поширені в багатьох областях, таких як астрономічні або медичні зображення, кольорова ангіографія очного дна для візуалізації сітківки [4]. Процес знімання зображення (або деконволюція) - це завдання реконструкції справжнього зображення із спостережуваного зашумованого зображення, відновлення важливих особливостей.

Математично згортка - це операція над двома функціями f і g , що виробляє функцію, яка розглядає кількість перекриття однієї функції при зміщенні її на іншу функцію [11]. При обробці зображення ядро складається з оператора зображення, що змінює значення пікселів зображення x залежно від значень сусідніх пікселів.

Нехай $f(t)$ і $w(t)$ - дві функції. Виток f і w , позначається як $w * f$, є функцією на $t \geq 0$, заданою через (1.14):

$$g(x) = w * f(x) = \int_0^x w(s)f(x - s)ds. \quad (1.14)$$

Це особливий вид інтегрального перетворення (1.15):

$$w * f(x) = \int_{-\infty}^{\infty} w(s)f(x - s)ds. \quad (1.15)$$

або в більш загальному випадку, якщо f і w є комплексними функціями на R^2 :

$$w * f(x) = \int_{R^n} w(s)f(x - s)ds. \quad (1.16)$$

Оскільки зображення зберігається як набір дискретних пікселів, нам потрібна дискретна згорнута форма (1.16). У дискретному вигляді інтеграл замінюється підсумовуванням, наприклад для 2-D функціональний інтеграл (1.17) можна записати так:

$$g(x, y) = w(x, y) * f(x, y) = \sum_{s=-\infty}^{\infty} \sum_{t=-\infty}^{\infty} w(s, t) f(x - s, y - t)ds. \quad (1.17)$$

Спостережуване розмите зображення можна записати як згортку справжнього зображення з функцією розведення точки або невідомою функцією ядра [15]. Нехай $z(x)$ (1.18) позначає отримане зображення, $h(x)$ - функція розмиття, $\eta(x)$ - шум, отриманий під час збору даних, $u(x)$ - справжнє зображення, яке слід відновити. Ми моделюємо розмите зображення як:

$$z(x) = [h * u](x) + \eta(x). \quad (1.18)$$

Існує три основні проблеми, що усунення розмиття.

1. Несліпа деконволюція: незважаючи на те, що функція поширення точок вважається відомою, це є рідкістю у реальних програмах. Мета - відновити справжній образ, який широко вивчається.

2. Напівсліпа деконволюція: інформація про ядро розмитості передбачається або відома.

Завдання полягає в оцінці правильних параметрів розмиття.

3. Деконволюція сліпого зображення: ядро зображення та розмиття є невідомими та завдання відновлення обох дуже складне [7].

Деконволюція сліпих зображень, як модель, представили Чан та Вонг у 1998 році. Як продовження моделі Вас і Каве, Чан-Вонг функціонал (1.19) представляється як як:

$$f(u, h) = \frac{1}{2} \|h(x) * u(x) - z(x)\|^2 + \alpha_1 \int_{\Omega} |\nabla u(x)|_{\beta} + \alpha_2 \int_{\Omega} |\nabla h(x)|_{\beta}, \quad (1.19)$$

де α_1 і α_2 - малі, не негативні параметри. Регуляризація(1.20) на u і h плавним наближенням TV функції:

$$\int_{\Omega} |\nabla h(x)|_{\beta} = \int_{\Omega} \sqrt{|\nabla h(x)|^2 + \beta^2}. \quad (1.20)$$

Функціонал $f(u, h)$ не є спільно опуклим відносно u та h . Відповідно, це (1.21) мінімізується з урахуванням наступних обмежень:

$$u(x) \geq 0, h(x) \geq 0, \int_{\Omega} h(x) dx = 1, h(x) = h(-x). \quad (1.21)$$

Для мінімізації функціональних рівнянь Ейлера-Лагранжа виводяться такі формули (1.22) та (1.23):

$$\frac{\partial}{\partial u} f(u, h): h^{\dagger}(x) * (h(x) * u(x) - z(x)) + \alpha_1 \nabla \left(\frac{u(x)}{|u(x)|_{\beta}} \right) = 0, \quad (1.22)$$

$$\frac{\partial}{\partial h} f(u, h): u^{\dagger}(x) * (u(x) * h(x) - h(x)) + \alpha_2 \nabla \left(\frac{h(x)}{|h(x)|_{\beta}} \right) = 0, \quad (1.23)$$

де $h^{\dagger}(x) = h(-x)$ і $u^{\dagger}(x) = u(-x)$ - примикання h і u відповідно. Описана схема почергової мінімізації, де обмеження введені вище накладаються на кожному етапі.

1.3.3 Реєстрація зображення

Реєстрація зображень є складним завданням вирівнювання двох зображень, щоб встановити відповідність між ознаками всередині них. Модерніший класифікує інтенсивність методів реєстрації двома визначеннями: непараметричні та параметричні. Останній ґрунтується на варіаційному підході, де метою є знайти поле деформації на фізичні процеси, такі як кривизна [18].

Альтернативні підходи передбачають основні методи, де певні місця визначають поле деформації. Можливо включити нелінійні методи реєстрації до варіативних моделей сегментації, але зазвичай ці підходи передбачають параметричні методи (тобто не варіаційні на основі).

Параметрична реєстрація зображення передбачає жорстку або афінну трансформацію, яка залежать від трьох та шести параметрів відповідно. Оскільки цей метод є особливим, коротко опишемо деякі деталі, пов'язані з афінною реєстрацією [1].

Шаблон і посилення, $T, R \in \Omega$, визначають зображення, які потрібно порівняти. Для $x \in \Omega$ позначають через $\varphi(x): \Omega \rightarrow \Omega$ невідоме перетворення координат, яке виробляє вирівнювання між посиленням, $R(x)$ і трансформованим шаблоном, $T(\varphi(x))$.

Перетворення є лінійним (1.24) і може бути визначено наступним чином:

$$\varphi(x) = \begin{bmatrix} a_1 & a_2 \\ a_4 & a_5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} a_3 \\ a_6 \end{bmatrix}, \quad (1.24)$$

задано шість параметрів $a = \{a_1, a_2, a_3, a_4, a_5, a_6\}$. Регламентована афінна реєстраційна модель(1.25) така:

$$\min_a \left\{ \eta R(a) + \frac{1}{2} \int_{\Omega} (T(\varphi(x)) - R(x))^2 dx \right\}, \quad (1.25)$$

де $\eta > 0$ - параметр зважування для регуляризації і сума квадрата

Термін відмінності (SSD) визначає схожість між посиленням і шаблоном. Розумний вибір регуляризації(1.26) параметру a :

$$R(a) = \frac{1}{2} ((1 - a_1)^2 + a_2^2 + a_3^2 + a_4^2 + (1 - a_5)^2 + a_6^2). \quad (1.26)$$

Можливі інші варіанти R , і хороші результати також можуть бути досягнуті без регуляризації для деяких прикладів [5]. Міра подібності, яку ми використовуємо, - це SSD(1.26), який ми визначаємо наступним чином:

$$D(a) = \frac{1}{2} \int_{\Omega} (T(\varphi(x)) - R(x))^2 dx, \quad (1.26)$$

Потім мінімізація кроку афінної реєстрації (1.27) дається як:

$$\min_{a \in \mathbb{R}^2} \{ J_{\eta}(a) = \eta r(a) + D(a) \}. \quad (1.27)$$

Ми вирішуємо це за допомогою оптимізаційного підходу. Використовуючи метод Гаусса Ньютона(1.28), ми можемо оновити a за допомогою пертурбації $\delta a^{(k)}$:

$$a^{(k+1)} = a^{(k)} + \delta a^{(k)}, \quad (1.28)$$

Пертурбація(1.29) Гаусса Ньютона $\delta a^{(k)}$ задається як:

$$H_{J_\eta}(a^{(k)})\delta a^{(k)} = -gJ_\eta(a^{(k)}), \quad (1.29)$$

де

$$H_{J_\eta}(a^{(k)}) = J^T(a^{(k)})J(a^{(k)}) + \eta H_R(a^{(k)}) \quad (1.30)$$

та

$$gJ_\eta(a^{(k)}) = \nabla_a D(a) + \eta \nabla_a R(a^{(k)}) \quad (1.31)$$

- апроксимований Гессіан(1.31) і градієнт J_η при $a^{(k)}$ та $H_R(a^{(k)})$ є градієнтом і Гессіана R при $a^{(k)}$ відповідно [8]. Наближений Гессіан використовується для розгляду з нелінійністю проблеми. Таким чином, обчислення похідних вищого порядку уникається.

1.4 Виокремлення об'єктів

Для того, щоб виокремити об'єкт на відеокадрах або на статичному зображенні необхідно використовувати спеціальні методи. Існує досить багато готових бібліотек та фреймворків, які можуть спростити розробнику цю задачу та значно заощадити час. Однак, для того, щоб краще розуміти як саме вони працюють, розглянемо з чого починається обробка зображення.

1.4.1 Сегментація

Сегментація зображення - це розподіл зображення на кілька областей спільних характеристик [5]. Різноманітні методи для цього складаються з мінімізації енергетичного функціоналу,

що веде до часткового диференціального рівняння на основі обчислення варіацій. Тоді рішення цього рівняння відповідає змістовному відображенню зображення.

Метою є знаходження замкнутого контуру Γ , який розділяє область $\Omega \in \mathbb{R}^2$

в субрегіони Ω_i , $i = 1, 2, \dots, N$. У цій тезі ми зупинимось на двофазних методах. Тобто ми хочемо розділити зображення на якесь значне подання переднього плану / фонового зображення, і $N = 2$.

Враховуючи дискретні спостережувані зображення, проблема задається в безперервній області, де мета - визначити рішення деякого рівняння, отриманого з функціональної енергії.

Цей процес може здатися не інтуїтивним, але варіативні методи виявилися дуже успішними з моменту роботи на семінарі Мумфорда і Шаха. Інші вагомі події з тих пір включають крайові результати методів, такі як наближення змії Касса, Віткіна та Терзопулоса та геодезична модель активних контурів Казелла, Кіммеля та Сапіро. Найважливіше, що успіхом цих підходів була розробка методів, які широко використовуються протягом останніх двадцяти років [17]. Це використовували Чан і Весе у впливовій моделі активних контурів без країв.

Поширеною темою такого підходу до сегментації є те, що проблеми не випуклі, це означає, що отримати глобальний мінімізатор часто неможливо.

Останні роботи, присвячені проблемі невивуклості, включають опуклі методи релаксації. Основна робота щодо сегментації в безперервній обстановці - це Чана, Еседоглу та Ніколова в 2006 [6]. Ця модель спрямована на пошук глобального мінімізації двофазної кусочно-постійної формули Мумфорда-Шаха у випадку відомих констант інтенсивності. Теоретична основа цієї роботи базується на роботі Странга.

Словом, опуклий метод релаксації складається з представлення кожної фази з двійковою функцією, $u \in \{0, 1\}$ та інтерфейсом неявно представленим місцем стрибка. Це можна побачити на рисунку 1.4, та

порівнювати з методом набору рівней (Levelset).



Рисунок 1.4—Ілюстрація представлення інтерфейсу в опуклій системі релаксації. Показує двійкову функцію u та відповідний контур, який неявно визначає інтерфейс.

У опуклої рамці релаксації є бінарне обмеження, а Γ задається процедурою порогоування параметра $\gamma \in (0, 1)$.

1.4.2 Мумфорд-шахський підхід

Важливою роботою в цій галузі є робота Мумфорда і Шаха у 1989 році. Це стосується кусково-плавного наближення вхідного зображення $z(x)$, парою (u, Γ) . Нехай Ω буде обмеженим доменом в R^2 і $z(x)$ - обмежена вимірювана функція, визначена на Ω .

Функція Мумфорд-Шах (1.32) визначається як:

$$E(u, \Gamma) = \nu H^{n-1}(\Gamma) + \mu^2 \int_{\Omega} (u - z)^2 dx + \int_{\Omega \setminus \Gamma} |\nabla u|^2 dx, \quad (1.32)$$

Функціонал містить $u \in C^1$ та два строки регулярності. Один накладає гладкість на u , а інший накладає регулярність на Γ з точки зору одновимірної міри Хаусдорфа. Пов'язаний функціонал у дискретних умовах, а не на безперервній області вперше були введені Геманом і Геманом [1], і вивчалися Блейком та Зіссерманом.

Обчислення мінімізаторів є дуже складним через нерегулярність крайового члена Γ . Також був запроваджений дуже схожий підхід в 1990 р. Амбросіо та Тортореллі, які наближали мінімізатори послідовністю простіших еліптичних

варіаційних задач. З тих пір було багато інших підходів, що запропонували методи пошуку мінімальної пари (u, Γ) на основі опуклих методів релаксації, на підходах до рівня.

Мамфорд і Шах також обговорюють обмеження E на кусочно-постійні функції u . Іншими словами, $u = c_k$ на кожному відкритому множині Ω_k .

Кусково-постійний функціонал Мумфорд-Шаха(1.33) дається як:

$$E_u(u, \Gamma) = \nu H^{n-1}(\Gamma) + \int_{\Omega_k} (u - c_k)^2 dx. \quad (1.33)$$

Можна довести, що E_0 є природним граничним функціоналом E як $\mu \rightarrow 0$. Цей зменшений реєстр також називається проблемою мінімального розділу. Він зазначає, що у дискретних умовах пов'язаних з моделлю Потта [12], яка широко вивчалася, нас особливо цікавить розподіл зображень на передній план та фон, тобто $N = 2$. Це відоме як двофазна кусочно-постійна функція (1.34) Мумфорда і Шаха і задається наступним чином:

$$E(\Gamma, c_1, c_2) = \nu |\Gamma| + \int_{\Omega_1} (z - c_1)^2 dx + \int_{\Omega_2} (z - c_2)^2 dx. \quad (1.34)$$

Вищенаведений функціонал є основою для значного обсягу важливої роботи в цій області досліджень.

1.4.3 Геодезичні контури

Ранній метод, заснований на PDE, для вилучення об'єктів на зображенні - це модель активних контурів (також відомий як Змії), запропонований Кассом, Віткіном та Терзополусом у 1988 році [13]. Підходи були дуже успішними і мали важливе значення в широкому діапазоні застосування, особливо в медичних візуалізаціях. У 1997 році Каселес, Кімел та Сапіро представили модель геодезичних активних контурів (1.35).

$$F_{GAG}(C(s)) = \int_0^{L(C)} g(|\nabla z(C(s))|) ds \quad (1.35)$$

де

$$g(\nabla z) = \frac{1}{1+\gamma|\nabla z|^2}. \quad (1.36)$$

Приклад цієї крайової функції для даного зображення можна побачити на рисунку 1.36. Ідея полягає в тому, що g невелика біля меж об'єкта, визначених стрибками інтенсивності z , і керується параметром γ . Рівняння Ейлера-Лагранжа функціонального і градієнтного спуску дають наступний PDE:

$$\frac{\partial c}{\partial t} = gk\vec{n} - (\nabla g \cdot \vec{n})\vec{n}, \quad (1.37)$$

де k - евклідова кривизна, а \vec{n} - одиничний нормальний вектор. Зважаючи на це, рівняння (1.37) можна переписати так:

$$\frac{\partial \Phi}{\partial x} = |\nabla \Phi| \left(\nabla \cdot \left(g \frac{\nabla \Phi}{|\nabla \Phi|} \right) + v g \right), \quad (1.38)$$

де Φ - функція Ліпшица (1.38), що представляє C як нульовий набір рівня. Тут константа v додається для залучення кривих до межі та збільшення швидкості еволюції.

Незважаючи на успіх геодезичних активних контурів, він обмежений у застосуванні по двом причинам [8]. По-перше, модель залежить від градієнта зображення. Це означає що зображення, які або містять шум, або мають межі, які не є чітко визначеними, не підходять для цього методу. По-друге, енергетичний функціонал не випуклий і тому дуже чутливий до ініціалізації.

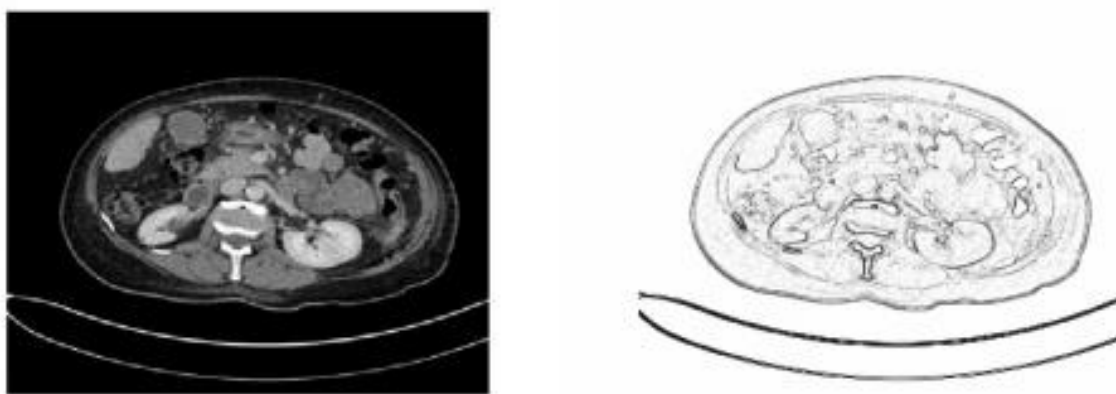


Рисунок 1.5 – Функція виявлення країв, $g(x)$ та $z(x)$.

1.4.4 Методи сегментації

Існує багато методів сегментації, тож розглянемо деякі з них. Сегментація зображення - це завдання розбити цифрове зображення на одне або декілька областей, що цікавлять розробника. Це основна проблема в галузі комп'ютерного зору, де використовуються різні технології, кожна з яких має переваги та недоліки. [2]

Сегментовані зображення часто потрібні в різних галузях. Припустимо, що в медицині він запроваджений в цифрових 3D-моделях для тих, хто цього потребує. Окрім того, сегментація використовується для розпізнаванні осіб, машинному зору тощо.

Наразі немає універсальних алгоритмів для сегментації, отже потрібно розробити спеціальні алгоритми, в яких параметри підбираються розробником, беручи до уваги конкретні умови. Окрім того, пошкодження зображень, наприклад зашумлення, неоднорідність, дуже важко брати до уваги у алгоритмах сегментації не враховуючи потреби розробника.[1] Точність сегментації впливає на успіх або невдачу процедур комп'ютерного аналізу.

«Snakes» - це метод двовимірного узагальнення 1D-мінімізуючих енергію сплайнів (1.39):

$$\varepsilon_{int} = \int \alpha(s) \|f_s(s)\|^2 + \beta(s) \|f_{ss}(s)\|^2 ds \quad (1.39)$$

де s – це довжина дуги вздовж кривої $f(s) = (x(s), y(s))$ та $\alpha(s)$ та $\beta(s)$ – функції зважування безперервності першого та другого порядку. Цю енергію можна дискретизувати методом вибіркового відбору початкової позиції кривої рівномірно впродовж її довжини, для того, щоб підрахувати енергію сплайнів (1.40):

$$E_{int} = \sum_i \alpha(i) \|f(i+1) - f(i)\|^2 / h^2 + \beta(i) \|\int(i+1) - 2\int f(i) + \int(i-1)\| \frac{1}{h^2} \quad (1.40)$$

де h – це розмір кроку, який може біти знятий, якщо ми повторно напишемо криву по довжині дуги після кожної ітерації.

Також метод «Snakes» мінімізує зовнішні потенціали. Сума декількох термінів є потенціал на основі зображення (1.41):

$$\varepsilon_{image} = w_{line}\varepsilon_{line} + w_{edge}\varepsilon_{edge} + w_{term}\varepsilon_{term} \quad (1.41)$$

де лінійний термін притягуєсплайн до темних хребтів. Однакв основному системи використовують тільки крайний термін, що може бути прямо пропорційним градієнтам зображення (1.42):

$$E_{edge} = \sum_i -||\nabla I(f(i))||^2 \quad (1.42)$$

або до згладженої версії зображення Лапласа (1.43):

$$E_{edge} = \sum_i -|(G_{\sigma} * \nabla^2 I)(f(i))|^2 \quad (1.43)$$

Метод набору рівнів – це бінарна сегментація зображення, що виконується методом фіксації рівня.

Такий спосіб був вперше запропонований Стенлі Ошером і Джеймсом Сетаном в 1980-і роки. Цей метод популярний в різнихсфрах, як наприклад комп'ютерна графіка, обчислювальна геометрія,оптимізація.

Метод набору рівня неявно перетворює поверхню або контур і використовує функцію більш високої розмірності, що має назву функція фіксації рівня. Перетворену поверхню або контур виражається як крива.[4]

Перевага цього методу в тому, що деякі топологічні зміни, як наприклад злиття або поділ поверхні або контуру, вирішуються неявно, що продемонстровано на рисунку 1.6.

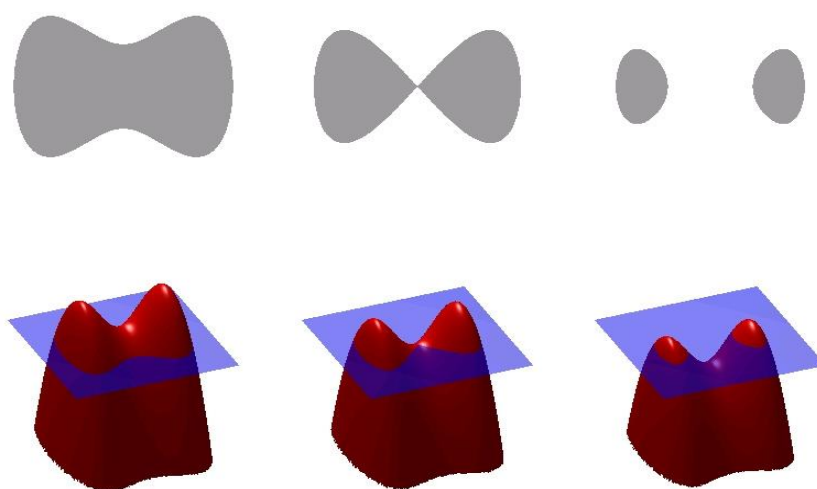


Рисунок 1.6—Взаємозв'язок між функцією набору рівня та деяким контуром

З рисунку 2.1 видно, що зміна поверхні розбиває контур.

Інший метод сегментації BlobColoring розроблений для визначення областей у зображенні, які децюрізні по властивостям, як наприклад колір або яскравість, порівнюючи з сусідніми областями.[6]

Можна сказати, що BLOB - це такий регіон зображення, у якому властивості можуть бути постійними або ж приблизно постійними.[9]

Алгоритм має наступні етапи.

Пороговість - це означає перетворення деяких вихідних зображень до двійкових зображень використовуючи поріг вихідного. Такі пороги зростають до maxThreshold . Можна зробити висновок, що першим порогом є minThreshold , другим - $\text{minThreshold} + \text{thresholdStep}$, третім - $\text{minThreshold} + 2 \times \text{thresholdStep}$ тощо.

Натупний етап – групування, що означає, що у кожному двійковому зображенні білі пікселі зв'язані [16].

Злиття –це означає, що центри бінарних блоків у деяких двійкових зображеннях обчислюються, а крапки, які можуть бути розташовані ближче, ніж $\text{minDistBetweenBlobs}$ можуть бути об'єднані.

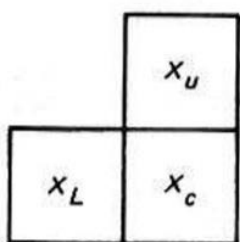


Рисунок 1.7–L-образний шаблон, що використовується для кольорового забарвлення

Коли завершується повне сканування зображення, тоді кольорові еквіваленти використовуються для того, щоб кожен об'єкт мав тільки один колір. Таке

ствердження рівне умовам $f(x_c) = f(x_u) = 1$. На рисунку 1.7 розглянуто шаблон, що використовується для кольорового забарвлення.

1.4.5 Алгоритм R-CNN

Замість того, щоб працювати над величезною кількістю регіонів, алгоритм R-CNN пропонує купу полів на зображенні і перевіряє, чи містить якийсь із цих скриньок який-небудь об'єкт.

R-CNN використовує селективний пошук для отримання цих полів із зображення (ці поля називаються регіонами) [9].

Нижче наведено короткий підсумок кроків, що виконуються в RCNN для виявлення об'єктів:

Спочатку ми беремо заздалегідь підготовлену згорнуту нейронну мережу. Потім ця модель перекаліфікується. Далі ми тренуємо останній рівень мережі на основі кількості класів, які потрібно виявити.

Третій крок - отримати регіон інтересів для кожного зображення. Потім ми переформуємо всі ці регіони, щоб вони відповідали розміру вводу CNN.

Отримавши регіони, ми навчаємо SVM для класифікації об'єктів та фону. Для кожного класу ми тренуємо один двійковий SVM [6]. Нарешті, ми тренуємо лінійну регресійну модель для створення більш чітких обмежувальних коробок для кожного ідентифікованого об'єкта на зображенні.

Поки ми бачили, як R-CNN може бути корисним для виявлення об'єктів. Але ця техніка має свої обмеження. Навчання моделі RCNN є дорогим і повільним завдяки наступним крокам.

По-перше це вилучення 2000 регіонів для кожного зображення на основі вибіркового пошуку.

По-друге, витяг функцій за допомогою CNN для кожної області зображення. Припустимо, у нас є N зображень, тоді кількість функцій CNN становитиме $N * 2000$.

Весь процес виявлення об'єктів за допомогою RCNN має три моделі: CNN для вилучення функцій, лінійний класифікатор SVM для ідентифікації об'єктів та регресійну модель для затулювання обмежувальних коробок. При поєднанні цих процесів RCNN дуже сповільнюється.

Висновки до розділу 1

У вищезазначеному розділі були розглянуті декілька методів і алгоритми, що використовуються для обробки та сегментації динамічного зображення.

Розглянуто основні способи підготовки зображення до подальшої обробки, та деякі типи нейронних мереж, що були використані для подальшого аналізу та порівняння методів виокремлення об'єкту на відеокадрах.

2 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ РОЗВ'ЯЗАННЯ ЗАДАЧ І ВИОКРЕМЛЕННЯ ОБ'ЄКТІВ

У цьому розділі ми розглянемо методи за допомогою яких були реалізована поставлена задача, а саме за допомогою бібліотеки OpenCV, що базується на каскадних методах визначення об'єкту на зображенні та фреймворку Tensorflow із використанням нейронних мереж типу R-CNN.

2.1 Бібліотека OpenCV

OpenCV (Open Source Computer Vision Library) - бібліотека програмного забезпечення з відкритим кодом та програмного забезпечення для машинного навчання. OpenCV була побудована для забезпечення загальної інфраструктури для програм комп'ютерного зору та для прискорення використання машинного сприйняття в комерційних продуктах.

Бібліотека має понад 2500 оптимізованих алгоритмів, що включає в себе вичерпний набір як класичних, так і сучасних алгоритмів комп'ютерного зору та машинного навчання [15]. Ці алгоритми можна використовувати для виявлення та розпізнавання облич, ідентифікації об'єктів, класифікації дій людини у відео, відстеження рухів камери, відстеження рухомих об'єктів, вилучення 3D-моделей об'єктів, отримання 3D-хмарних точок із стереокамер, скріплення зображень разом для отримання високої роздільної здатності зображення цілої сцени, пошук подібних зображень з баз даних зображень, виправлення червоних очей на зображеннях, зроблених за допомогою спалаху, відслідковування руху очей, розпізнавання краєвидів тощо. У OpenCV є понад 47 тисячна користувачка спільнота та приблизно 18 мільйонів завантажень. Бібліотека широко використовується у компаніях, наукових групах та урядових органах.

Поряд з налагодженими компаніями, такими як Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota, які використовують бібліотеку, існує безліч стартапів, таких як Applied Minds, VideoSurf і Zeitera, які широко використовують OpenCV. Розгорнутий OpenCV використовується для виявлення вторгнень у відео спостереження в Ізраїлі, моніторингу шахтного обладнання в Китаї, допомоги роботам у навігації та підборі предметів у Willow Garage, виявлення аварій, запуску інтерактивного мистецтва в Іспанії та Нью-Йорку, перевірка злітно-посадкових смуг на сміття в Туреччині, огляд етикетки на продуктах на заводах по всьому світу для швидкого виявлення обличчя в Японії [6].

Він має інтерфейси C ++, Python, Java та MATLAB та підтримує Windows, Linux, Android та Mac OS. OpenCV в основному схиляється до додатків зору в режимі реального часу та використовує інструкції MMX та SSE, коли вони доступні. Зараз активно розробляються повнофункціональні інтерфейси CUDA та OpenCL. Існує понад 500 алгоритмів і приблизно в 10 разів більше функцій, які складають або підтримують ці алгоритми. OpenCV написано на мові C ++ і має шаблонний інтерфейс, який легко працює з контейнерами STL.

Каскад Хаара - алгоритм виявлення об'єктів на зображенні чи відео, який використовується для ідентифікації об'єктів у зображенні чи відео та заснований на концепції особливостей, запропонованих Полом Віолою та Майклом Джонсом у їхній роботі "Швидке виявлення об'єктів за допомогою посиленого каскаду простих функцій" в 2001 році.

Це підхід на основі машинного навчання, де функція каскаду навчається з безлічі позитивних і негативних образів [2]. Потім він використовується для виявлення об'єктів на інших зображеннях.

Алгоритм має чотири етапи: вибір функції Хаар, створення інтегральних образів, навчання Адабоосту, каскадні класифікатори.

Він добре відомий тим, що вміє виявляти обличчя та частини тіла на зображенні, але може бути навчений ідентифікувати практично будь-який об'єкт.

Розглянемо обличчя як приклад. Спочатку алгоритму потрібно багато позитивних зображень обличчя та негативних зображень без граней для підготовки класифікатора. Тоді нам потрібно витягти з нього функції.

Перший крок - зібрати характеристики Хаара [2]. Функція Хаара враховує сусідні прямокутні області у визначеному місці у вікні виявлення, підсумовує інтенсивність пікселів у кожній області та обчислює різницю між цими сумами, як показано на рисунку 2.1.

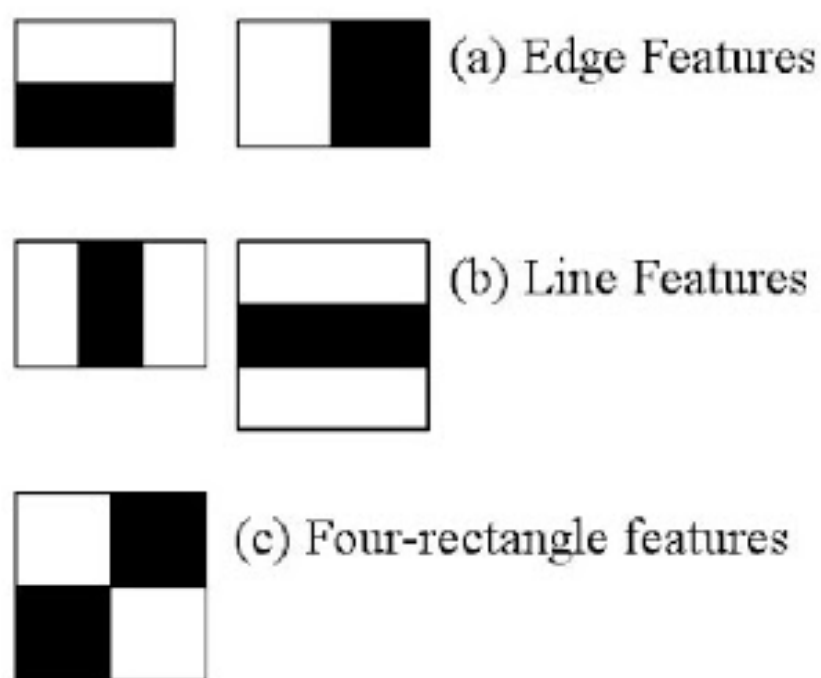


Рисунок 2.1 - Ознаки Хаара

Інтегральні зображення використовуються для того, щоб зробити це дуже швидко.[1]

Але серед усіх цих рисунків, більшість з них не мають значення. Наприклад, розглянемо зображення нижче. У верхньому рядку показано дві особливості. Перша обрана особливість, здається, фокусується на властивості того, що область очей часто темніша, ніж область носа та щік. Друга обрана особливість покладається на

властивість те, що очі темніші за мостик носа. Але ті ж вікна, що застосовуються на щоках або будь-якому іншому місці, не мають значення, як показано на рисунку 2.2.

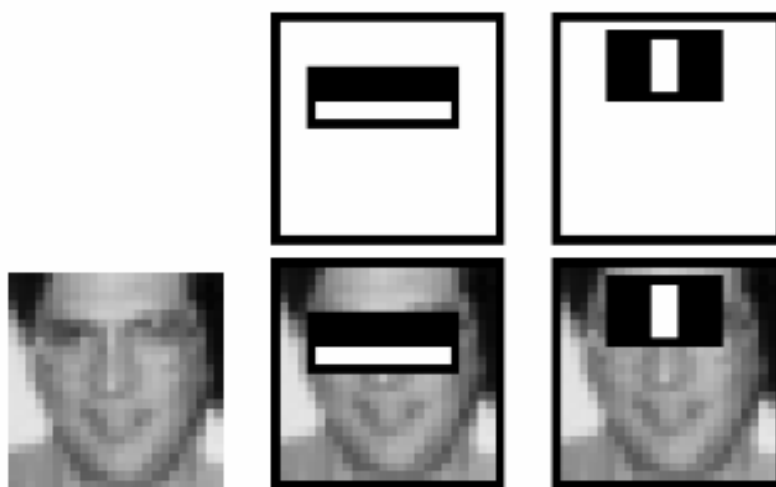


Рисунок 2.2 – Приклад розпізнавання очей за характеристиками Хаара

Тож як з 160000+ функцій вибрати найкращі функції? Це здійснюється за допомогою концепції під назвою Adaboost, яка обирає найкращі функції та навчає класифікатори, які ними користуються. Цей алгоритм будує "сильний" класифікатор як лінійну комбінацію зважених простих "слабких" класифікаторів. Процес полягає в наступному.

Під час фази виявлення вікна цільового розміру переміщується над вхідним зображенням, і для кожного підрозділу зображення обчислюються характеристики Хаара. Оскільки кожна характеристика Хаара [12] є лише "слабким класифікатором" (якість її виявлення трохи краща, ніж випадкові здогадки), для опису об'єкта з достатньою точністю необхідна велика кількість характеристик Хаара, і тому вони організовані в каскадні класифікатори для формування сильного класифікатора.

Каскадний класифікатор складається з набору етапів, де кожен етап - це ансамбль слабких учнів. Слабкі учні - це прості класифікатори, які називаються пнями [1]. Кожен етап тренується за допомогою техніки, яка називається прискоренням. Стимулювання забезпечує можливість підготовки високоточного класифікатора, приймаючи середньозважене рішення, прийняте слабкими учнями.

Кожен етап класифікатора позначає область, визначену поточним розташуванням розсувного вікна, або позитивним, або негативним. Позитивне означає, що об'єкт знайдено, а негативне означає, що об'єктів не знайдено. Якщо мітка негативна, класифікація цього регіону завершена [7], і детектор ковзає вікно до наступного місця. Якщо мітка позитивна, класифікатор переходить область до наступного етапу. Детектор повідомляє про об'єкт, знайдений у поточному місці вікна, коли заключний етап класифікує область як позитивну.

Етапи призначені для якнайшвидшого відхилення негативних зразків. Припущення полягає в тому, що переважна більшість вікон не містять об'єкта, що шукається.

Щоб добре працювати, кожен етап каскаду повинен мати низький показник помилкових негативних ефектів. Якщо етап неправильно позначає об'єкт як негативний, класифікація припиняється, і вже неможливо виправити помилку. Однак кожен етап може мати високу помилкову позитивну швидкість. [8] Навіть якщо детектор неправильно позначає не об'єкт як позитивний, ви можете виправити помилку на наступних етапах. Додавання більше етапів знижує загальну помилкову позитивну ставку, але також знижує загальну справжню позитивну ставку.

Навчання каскадного класифікатора вимагає набору позитивних зразків та набору негативних зображень. Ви повинні надати набір позитивних зображень із зацікавленими регіонами, які використовуються як позитивні зразки. Ви можете використовувати Image Labeler [11] для позначення об'єктів, що цікавлять, обмежуючи полями. Image Labeler виводить таблицю для використання для позитивних зразків. Щоб досягти прийнятної точності детектора, необхідно встановити кількість етапів, тип функції та інші параметри функції, як показано на рисунку 2.3.

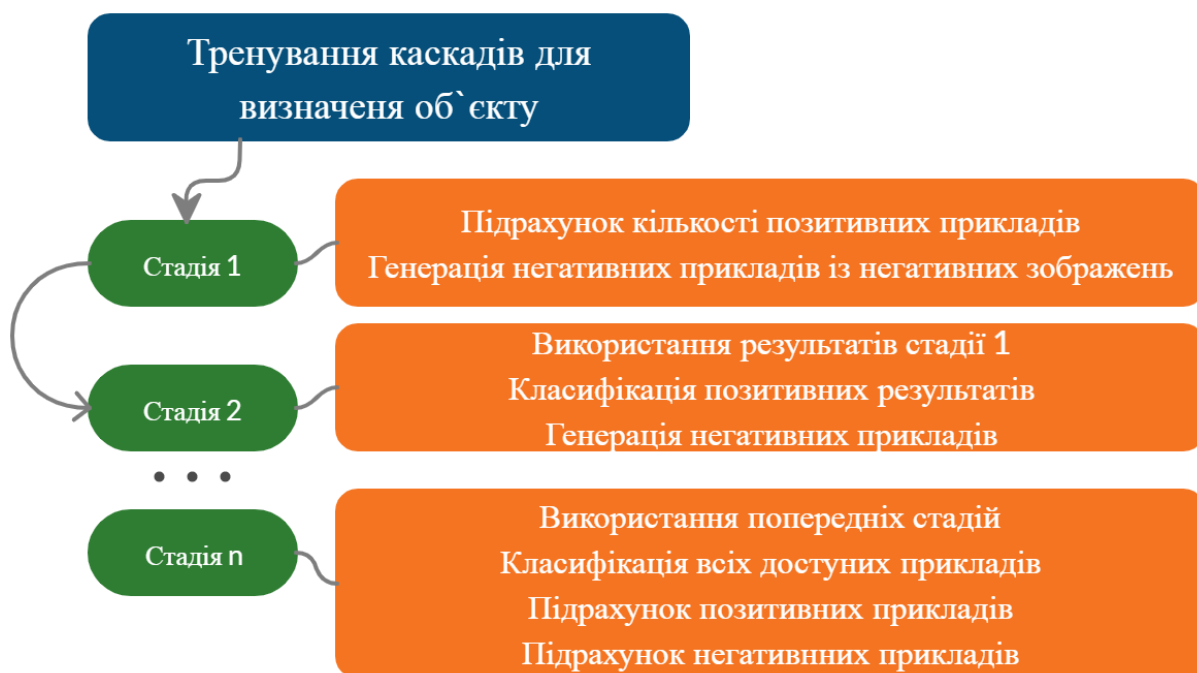


Рисунок 2.3 – Схема тренування каскадів

Також необхідно надати набір негативних зображень, з яких функція автоматично генерує негативні зразки.

2.2 Фреймворк Tensorflow

Tensorflow - це безкоштовна бібліотека програмного забезпечення з відкритим кодом для потоку даних та диференційованого програмування для різних завдань. Вона може тренувати та запускати глибокі нейронні мережі для рукописної класифікації цифр, розпізнавання зображень, вбудовування слів, періодичних нейронних мереж, моделей послідовності в послідовності для машинного перекладу, обробки природних мов та моделювання на основі PDE [2] (часткове диференціальне рівняння). Найкраще, що TensorFlow підтримує прогнозування виробництва в масштабі, з тими ж моделями, що використовуються для тренувань.

TensorFlow - це символічна математична бібліотека, а також використовується для машинного навчання, таких як нейронні мережі.

Нейронна мережа - це тип машинного навчання, який моделюється на людського мозку. Це створює штучну нейронну мережу, яка за допомогою алгоритму дозволяє комп'ютеру вчитися, включаючи нові дані.

Хоча в наші дні існує багато алгоритмів штучного інтелекту, нейронні мережі здатні виконувати те, що називалося глибоким навчанням [10]. Хоча основною одиницею мозку є нейрон, основним складовим елементом штучної нейронної мережі є перцептрон, який здійснює просту обробку сигналу, і вони потім з'єднуються у велику сітчасту мережу.

Комп'ютер з нейронною мережею вчать виконувати завдання, аналізуючи приклади тренувань, які були попередньо позначені.

Поширеним прикладом завдання для нейронної мережі з використанням глибокого навчання є завдання розпізнавання об'єктів, де нейронна мережа представлена великою кількістю об'єктів певного типу, наприклад, кішка, або дорожній знак, і комп'ютер, аналізуючи повторювані візерунки у представлених зображеннях, вчиться класифікувати нові зображення.

Як навчаються нейронні мережі? На відміну від інших алгоритмів, нейронні мережі з їх глибоким навчанням не можуть бути запрограмовані безпосередньо для виконання завдання. Вони повинні, як і мозок дитини, що розвивається, вивчати інформацію. Стратегії навчання здійснюються трьома методами:

Контрольоване навчання - ця стратегія навчання є найпростішою, оскільки існує мічений набір даних, через які проходить комп'ютер, і алгоритм змінюється, поки він не зможе обробити набір даних, щоб отримати бажаний результат.

Непідконтрольне навчання - ця стратегія застосовується у випадках, коли немає міченого набору даних для навчання [18]. Нейронна мережа аналізує набір даних, а потім функція витрат повідомляє нейронній мережі, наскільки далеко від цілі вона була. Потім нейронна мережа коригується для підвищення точності алгоритму.

Підсилене навчання - у цьому алгоритмі нейронна мережа зміцнюється за позитивні результати і карається за негативний результат, це змушує нейронну мережу вчитися з часом.

Хоча нейронні мережі, безумовно, представляють потужну сучасну комп'ютерну технологію, ідея сходить до 1943 року, з двома дослідниками Чиказького університету, Уорреном Маккаллоу, нейрофізіологом та математиком Вальтером Пітсом.

Їх стаття "Логічне обчислення ідей, що мають постійну нервову активність" вперше була опублікована в журналі "Теорія мозку" [16], який популяризував теорію про те, що активація нейрона є основною одиницею мозкової діяльності.

Однак ця стаття мала більше спільного з розвитком когнітивних теорій у той час, і двоє дослідників переїхали до MIT в 1952 році, щоб створити перший відділ когнітивних наук.

Нейронні мережі в 1950-х роках були родючою зоною для досліджень комп'ютерних нейронних мереж, включаючи Perceptron [5], який здійснив візуальне розпізнавання образів на основі складеного ока мухи. У 1959 році два дослідники Стенфордського університету розробили MADALINE (Multiple ADaptive LINEar Elements), нейронна мережа вийшла за межі теоретичної та вирішила актуальну проблему.

MADALINE була спеціально застосована для зменшення кількості відлуння по телефонній лінії, для покращення якості голосу та була такою успішною, що залишається в комерційному використанні до сучасних часів [2].

Незважаючи на початковий ентузіазм у штучних нейронних мережах, примітна книга 1969 року з MIT, Perceptrons: In Introduction to Computation Geometry, пригнічувала це.

Автори висловили скептицизм щодо штучних нейронних мереж, і що це, ймовірно, тупик у пошуках справжнього штучного інтелекту. Це значно притупило цю сферу для досліджень протягом 1970-х років. Незважаючи на це, деякі зусилля продовжувались, і в 1975 р. була розроблена перша багатошарова мережа, що

проклала шлях для подальшого розвитку нейронних мереж, досягнення, яке деякі вважали неможливим менш ніж за десятиліття до цього.

Інтерес у 1982 році значно відновився до нейронних мереж, коли Джон Хопфілд, професор Принстонського університету, винайшов асоціативну нейронну мережу; нововведення полягало в тому, що дані можуть подорожувати в обидва боки, оскільки раніше вони були лише однонаправленими, також відомий своїм винахідником як мережа Хопфілда. Ідучи вперед, штучні нейронні мережі користуються широкою популярністю та зростанням.

Глибоке навчання стає особливо захоплюючим зараз, оскільки ми маємо більше обсягу даних та великих нейронних мереж, з якими можна працювати.

Більше того, продуктивність нейронних мереж покращується [3], оскільки вони збільшуються та працюють із більшою кількістю даних, на відміну від інших алгоритмів машинного навчання.

Нейронна мережа має велику кількість процесорів. Ці процесори працюють паралельно, але розташовані як рівні.

Перший рівень отримує вихідний вхід, аналогічний тому, як зоровий нерв отримує початкову інформацію у людини [8].

Кожен наступний рівень після цього отримує вхід від рівня перед ним, а потім передає свій вихід на рівень після нього. Останній ярус обробляє кінцевий результат.

Невеликі вузли складають кожен ярус. Вузли сильно взаємопов'язані з вузлами в ярусі до і після. Кожен вузол нейронної мережі має власну сферу знань, включаючи правила, за якими він був запрограмований, та правила, які він засвоїв сам.

Ключовим фактором ефективності нейронних мереж є те, що вони надзвичайно пристосовані та навчаються дуже швидко. Кожен вузол зважає важливість входу, який він отримує від вузлів перед ним [7]. Вхід, який найбільше сприяє правильному виходу, отримує найбільшу вагу.

Розглянемо які існують різні типи нейронних мереж. Різні типи нейронних мереж використовують різні принципи у визначенні власних правил.

Існує багато видів штучних нейронних мереж, кожна зі своїми унікальними здібностями.

Нейронна мережа, що рухається - штучний нейрон.

Це один з найпростіших видів штучних нейронних мереж. У цій нейронній мережі дані проходять через різні вхідні вузли, поки не досягнуть вихідного вузла.

Іншими словами, дані рухаються лише в одному напрямку від першого ярусу до тих пір, поки вони не досягнуть вихідного вузла[11]. Це також відоме як передня розповсюджена хвиля, яка зазвичай досягається за допомогою класифікаційної функції активації.[5]

На відміну від більш складних типів нейронних мереж, немає зворотного розповсюдження і дані рухаються лише в одному напрямку. Нейронна мережа, що подається, може мати один шар або вона може мати приховані шари.

У нейронній мережі, що подається, обчислюється сума продуктів входів та їх ваги. Потім подається на вихід. Ось приклад одношарової подавальної нейронної мережі, як показано на рисунку 2.4.

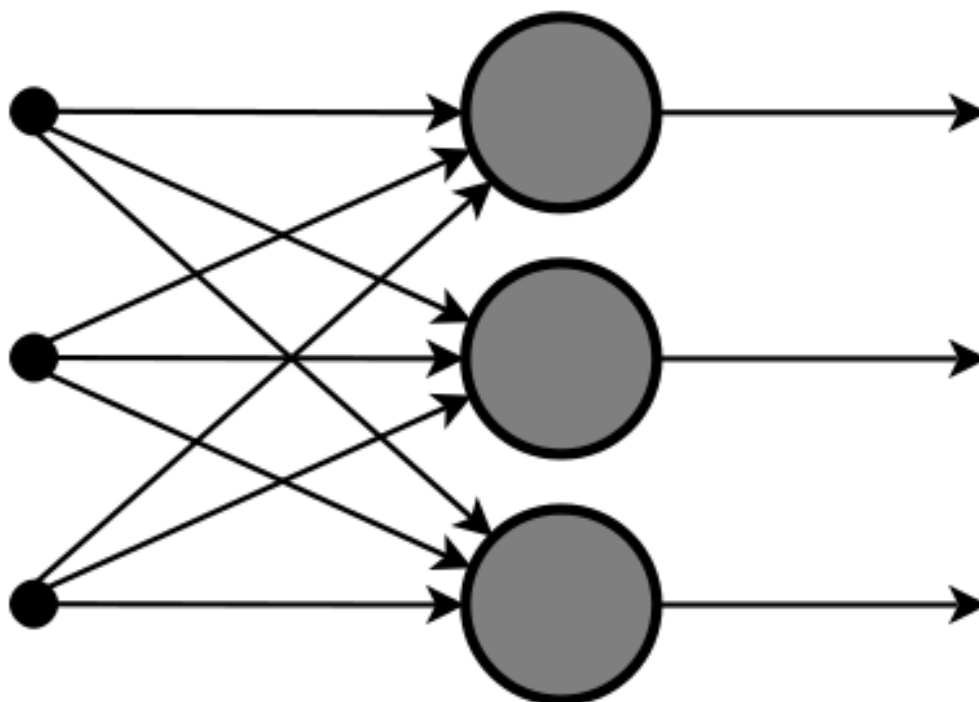


Рисунок 2.4 – Нейронна мережа, що рухається - штучний нейрон

Нейронні мережі, що рухаються, застосовуються в таких технологіях, як розпізнавання обличчя та комп'ютерний зір. Це тому, що цільові класи в цих додатках важко класифікувати.

Проста нейронна мережа, що подається вперед, обладнана для обробки даних, що містять багато шуму. Нейронні мережі, що подаються, також досить прості в обслуговуванні.[9]

Нейронна мережа функції радіальної основи.

Функція радіальної основи розглядає відстань будь-якої точки відносно центру. Такі нейронні мережі мають два шари. У внутрішньому шарі функції поєднуються з функцією радіальної основи.

Тоді вихід цих особливостей враховується при обчисленні того ж виходу на наступному часовому кроці. Ось схема, яка представляє радіальну основу функції нейронної мережі, як показано на рисунку 2.5.

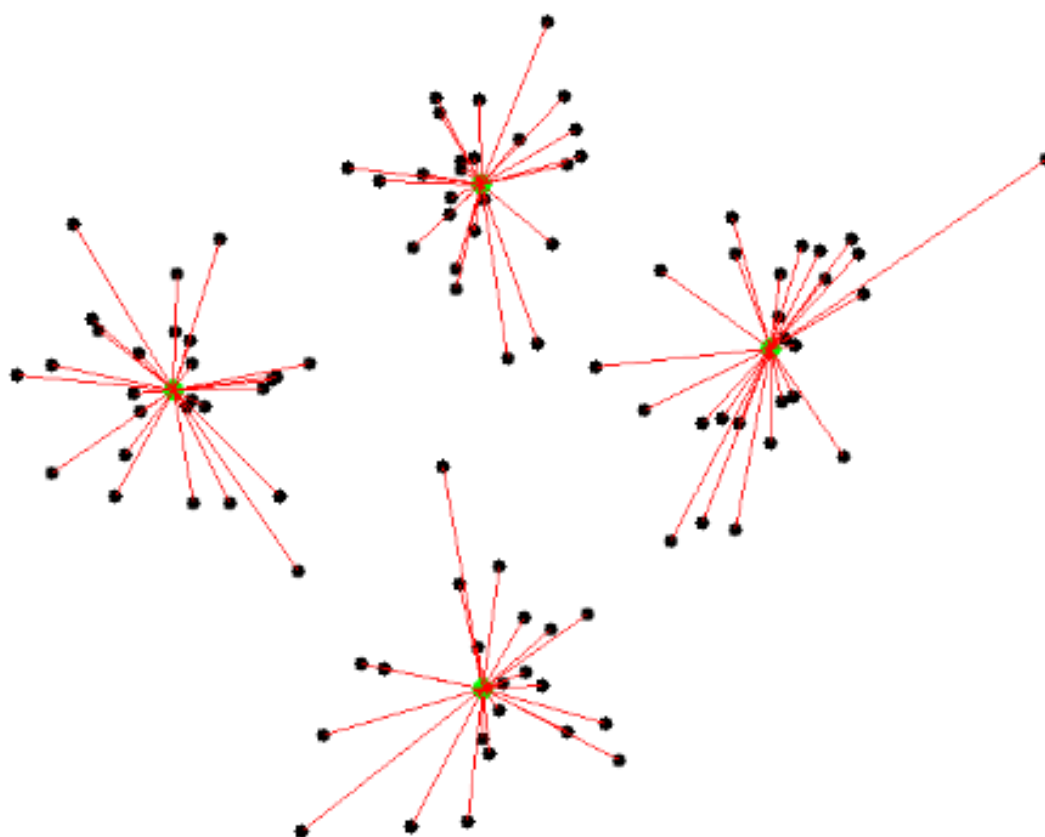


Рисунок 2.5 – Нейронна мережа функції радіальної основи

Нейронна мережа радіальної основи функціонує широко в системах відновлення енергії. В останні десятиліття енергосистеми стають більшими та складнішими.[17]

Це збільшує ризик затемнення. Ця нейронна мережа використовується в системах відновлення енергії з метою відновлення живлення в найкоротші терміни.

Багатошаровий перцептрон має три або більше шарів. Він використовується для класифікації даних, які неможливо розділити лінійно.[6]

Це тип штучної нейронної мережі, яка повністю пов'язана. Це відбувається тому, що кожен окремий вузол у шарі з'єднаний з кожним вузлом у наступному шарі.

Багатошаровий перцептрон використовує нелінійну функцію активації (головним чином гіперболічну дотичну або логістичну функцію). Ось як виглядає багатошаровий перцептрон, як показано на рисунку 2.6.

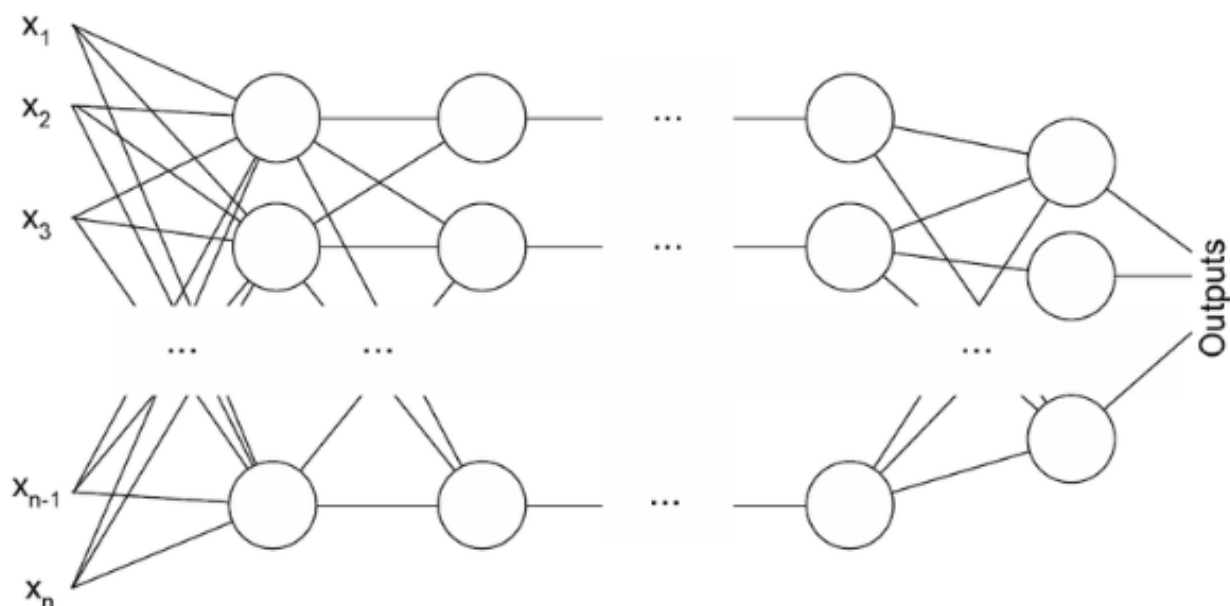


Рисунок 2.6 – Багатошаровий перцептрон

Цей тип нейронної мережі широко застосовується в технологіях розпізнавання мовлення та технологіях машинного перекладу.

Конволюційна нейронна мережа (CNN) використовує варіацію багатошарових перцептронів. CNN містить один або декілька згорткових шарів. Ці шари можуть бути повністю пов'язані між собою або об'єднані.

Перш ніж передати результат наступному шару, згортковий шар використовує згорнуту операцію на вході.

Завдяки цій згортковій роботі мережа може бути набагато глибшою, але зі значно меншою кількістю параметрів.

Завдяки цій здатності конвертовані нейронні мережі демонструють дуже ефективні результати в розпізнаванні зображень та відео та природній обробці мови.

Конволюційні нейронні мережі також показують чудові результати в семантичному розборі та виявленні парафрази [1]. Вони також застосовуються при обробці сигналів та класифікації зображень.

CNN також використовуються для аналізу зображень та розпізнавання в сільському господарстві, де погодні особливості витягуються із супутників типу LSAT для прогнозування росту та врожаю земельної ділянки. Ось зображення, як виглядає конволюційна нейронна мережа, як показано на рисунку 2.7.

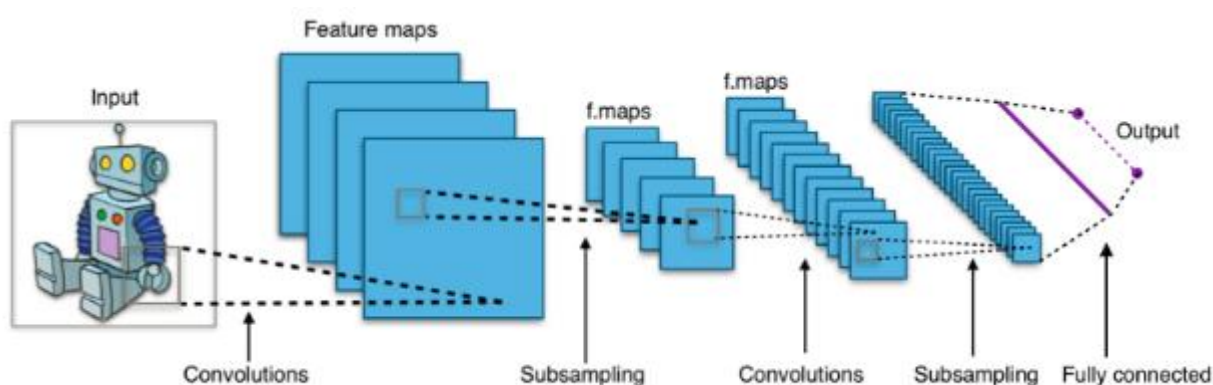


Рисунок 2.7 – Конволюційна нейронна мережа

Рекурентна нейронна мережа (RNN)- це тип штучної нейронної мережі, в якій вихід певного шару зберігається і подається назад на вхід. Це допомагає передбачити результат шару.

Перший шар формується так само, як і в мережі подачі. Тобто з добутком суми ваг та ознак. Однак у наступних шарах починається повторюваний процес нейронної мережі [7].

Від кожного часового кроку до наступного кожен вузол запам'ятовуватиме інформацію, яку він мав у попередньому часовому кроці.

Іншими словами, кожен вузол виконує функцію комірки пам'яті під час обчислення та виконання операцій. Нейронна мережа починається з переднього розповсюдження, як зазвичай, але запам'ятовує інформацію, яку, можливо, потрібно буде використовувати згодом.

Якщо прогноз неправильний, система самостійно навчається та працює над правильним прогнозуванням під час розмноження [9]. Цей тип нейронної мережі дуже ефективний в технології перетворення тексту в мовлення. Ось як виглядає періодична нейронна мережа, як показано на рисунку 2.8.

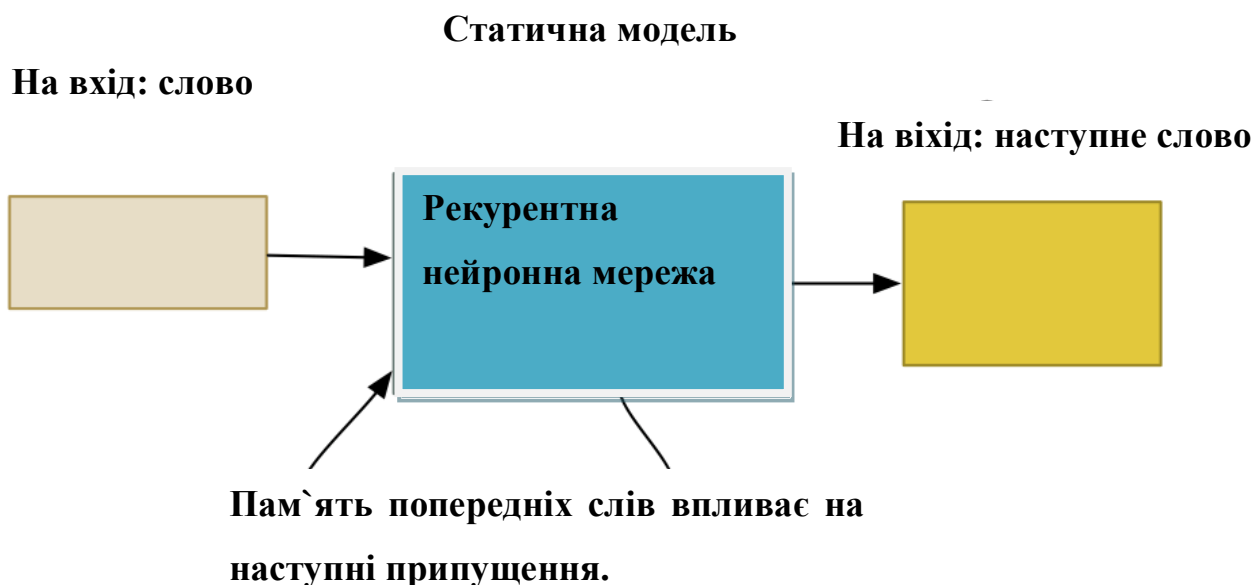


Рисунок 2.8 – Рекурентна нейронна мережа

Модульна нейронна мережа має ряд різних мереж, які функціонують незалежно та виконують підзадачі. Різні мережі насправді не взаємодіють між собою та не передають сигнал під час обчислення. Вони працюють незалежно щодо досягнення результату [4].

В результаті великий і складний обчислювальний процес можна зробити значно швидше, розбивши його на незалежні компоненти. Швидкість обчислення зростає, оскільки мережі не взаємодіють і навіть не з'єднуються між собою. Ось наочне зображення модульної нейронної мережі, як показано на рисунку 2.9.

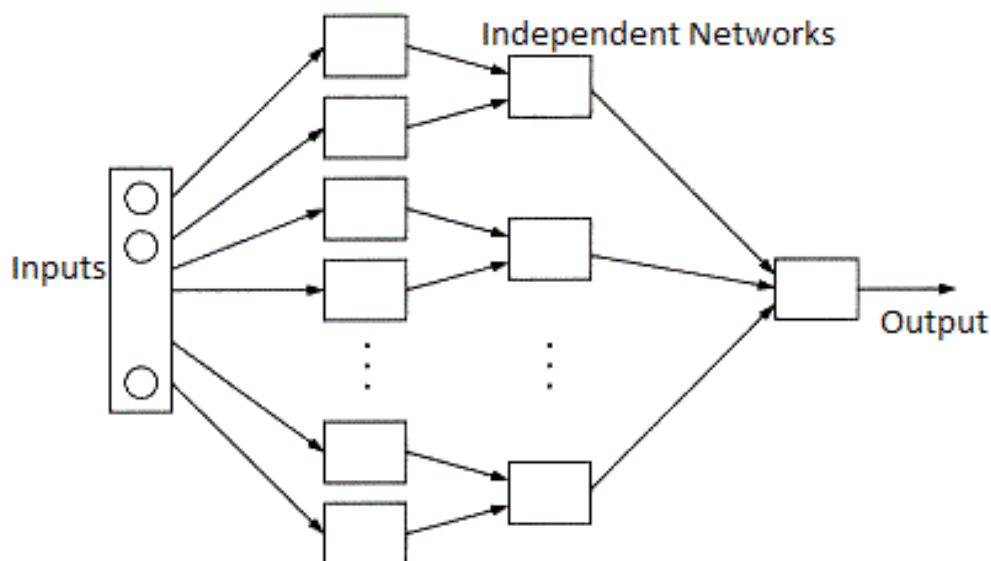


Рисунок 2.9 – нейронна мережа

Модель послідовності до послідовності складається з двох повторюваних нейронних мереж. Є кодер, який обробляє вхід, і декодер, який обробляє вихід. Кодер і декодер можуть використовувати однакові або різні параметри. Ця модель застосовується в тих випадках, коли довжина вхідних даних не така, як довжина вихідних даних.

Моделі послідовність до послідовності застосовуються в основному в чатах, машинному перекладі та системах відповідей на питання.

Розглянемо реальні місця використання нейронних мереж. Розпізнавання рукописного тексту - це приклад реальної світової проблеми, до якої можна звернутися через штучну нейронну мережу[11]. Завдання полягає в тому, що люди можуть розпізнавати почерк інтуїтивно, але завдання для комп'ютерів полягає в тому, що почерк кожної людини є унікальним, з різними стилями та навіть різним інтервалом між літерами, що ускладнює послідовне розпізнавання.

Наприклад, першу букву з великої літери А можна описати як три прямі, де дві зустрічаються на вершині вгорі, а третя знаходиться поперек двох інших на півдорозі вниз, і має сенс для людей, але це складно виразити це в комп'ютерному алгоритмі.[3]

Приймаючи штучну нейронну мережу, комп'ютер подає навчальні приклади відомих рукописних символів, які раніше були позначені, якій букві чи цифрі вони відповідають, і за допомогою алгоритму комп'ютер потім вчиться розпізнавати кожен символ, тож як набір символів збільшується, збільшується і точність. Розпізнавання рукописного тексту має різноманітні програми, такі як автоматичне читання адреси на листах на поштовій службі, зменшення шахрайства в банку при чеках, для введення символів для обчислень на основі ручки. [1]

Машинне навчання - складна дисципліна. Але впровадження моделей машинного навчання набагато менш складне, ніж раніше, завдяки машинним структурам навчання - таким як Google TensorFlow, що полегшує процес отримання даних, навчальних моделей, подання прогнозів та уточнення майбутніх результатів.

Створений командою Google Brain, TensorFlow - це бібліотека з відкритим кодом для чисельних обчислень та масштабованого машинного навчання. TensorFlow поєднує в собі ряд моделей та алгоритмів машинного навчання та глибокого навчання (також нейронних мереж). [8]

Він використовує Python для надання зручного інтерфейсного API для побудови програм із основою, виконуючи ці програми у високопродуктивних C++.

Розглянемо як працює TensorFlow. TensorFlow дозволяє розробникам створювати графіки потоків даних - структури, що описують, як дані переміщуються через графік або серію вузлів обробки.

Кожен вузол у графі представляє математичну операцію, і кожне з'єднання або край між вузлами є багатовимірним масивом даних або тензором.

TensorFlow забезпечує все це для програміста за допомогою мови Python. Python легко вивчити та працювати з ним, а також ця мова програмування пропонує зручні способи, за допомогою яких абстракції високого рівня можна поєднати разом. Вузли і тензори в TensorFlow є об'єктами Python, а програми TensorFlow - самі програми Python.[12]

Однак фактичні математичні операції в Python не виконуються. Бібліотеки перетворень, доступних через TensorFlow, записуються як високопродуктивні

бінарні файли C ++. Python просто спрямовує трафік між фрагментами і забезпечує абстракції програмування високого рівня, щоб з'єднати їх разом.

Програми TensorFlow можна запускати на локальній машині, кластері в хмарі, пристроях iOS та Android, [18] процесорах чи графічних процесорах. Якщо ви використовуєте власну хмару Google, для подальшого прискорення можна запустити TensorFlow на спеціальному технологічному блоку TensorFlow (TPU) Google. Отримані в результаті моделі, створені TensorFlow, можуть бути розгорнуті на більшості будь-яких пристроїв, де вони будуть використовуватися для подання прогнозів.

TensorFlow 2.0, випущений у жовтні 2019 року, багато в чому оновив свою основу за відгуками користувачів, щоб полегшити роботу (наприклад, за допомогою відносно простого API Keras для навчання моделей) та більш ефективної роботи. Розподілений тренінг простіше проводити завдяки новому API, а підтримка TensorFlow Lite [6] дозволяє розгорнути моделі на великій кількості різноманітних платформ.

Однак код, написаний для більш ранніх версій TensorFlow, повинен бути переписаний (іноді лише незначно, іноді значно), щоб максимально скористатися новими можливостями TensorFlow 2.0.[5]

Висновки до розділу 2

У цьому розділі був розглянутий каскадний метод виокремлення об'єкту на зображення та бібліотека OpenCV, що використовує цей метод.

Також у розділі був розглянутий фреймворк Tensorflow та його основні властивості. За його допомогою стало значно простіше розробляти програми машинного навчання, таких як нейронні мережі.

Також було розглянуто багато типів штучних нейронних мереж, які діють по-різному для досягнення різних результатів. На відміну від традиційних алгоритмів машинного навчання, які мають тенденцію до застою після певного моменту, нейронні мережі мають можливість по-справжньому зростати з більшою кількістю даних та більшою кількістю використання.

Ось чому багато експертів вважають, що різні типи нейронних мереж будуть основою, на якій буде будуватися Штучний інтелект нового покоління.

3 СХЕМА РОЗВ'ЯЗАННЯ ЗАДАЧІ

Для розробки схеми етапів розв'язання задачі з виокремлення об'єктів на відеокадрах були використані бібліотека OpenCV та фреймворк Tensorflow із використанням нейронних мереж.

Розроблені схеми етапів було застосовано до програмних забезпечень побудованих з використанням цих технологій. Були розглянуті основні модулі бібліотеки OpenCV та схему розв'язання задачі за допомогою Tensorflow.[14]

3.1 Схеми розв'язання на основі OpenCV

Розглянемо схему розв'язання задачі з використанням бібліотеки OpenCV, як показано на рисунку 3.1.

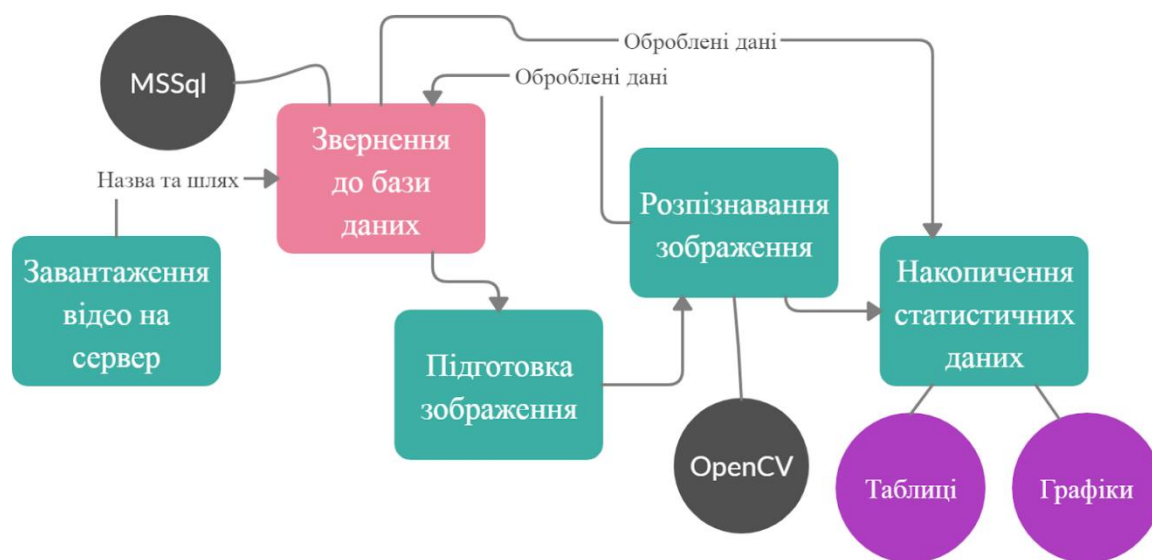


Рисунок 3.1 – Схему розв'язання задачі з використанням бібліотеки OpenCV

Спочатку необхідно завантажити відео на сервер. [9] Для зберігання даних використовується MsSql. Потім потрібно підготувати зображення, а саме прибрати шуми, розбити відео на кадри.

Кожен кадр оброблюється за допомогою бібліотеки OpenCV та в залежності від заданого класифікатора визначає об'єкт на зображенні. [8]

На виході обробляємо дані та виводимо їх у вигляді графіків або таблиць. Основні модулі OpenCV розглянуті нижче у вигляді таблиці 3.1.

Таблиця 3.1 – Модулі бібліотеки OpenCV

№	Назва	Призначення функції
1.	opencv_core	Базові структури, обчислення (математичні функції, генерація псевдовипадкових чисел, DFT, DCT, ввід/вивід в XML і т.д.)
2.	opencv_highgui	Простий UI, завантаження/збереження зображень та відео
3.	opencv_ml	Методи та моделі машинного навчання (SVM, дерева прийняття рішень і т. д.)
4.	opencv_features2d	Різні дескриптори (SURF)
5.	opencv_video	Аналіз пересування та відстежування об'єктів (оптичний потік, шаблони пересування, усунення фону).
6.	opencv_objdetect	Виявлення об'єктів на зображенні (вейвлети Хаара, HOG і т. д.)
7.	opencv_calib3d	Калібровка камери, пошук стерео-відповідності та елементи обробки трехмірних даних
8.	opencv_flann	Бібліотека швидкого пошуку найближчих сусідів (FLANN).

Приклади розпізнавання різних об'єктів за допомогою OpenCV можна розглянути на рисунках 3.2 та 3.3.

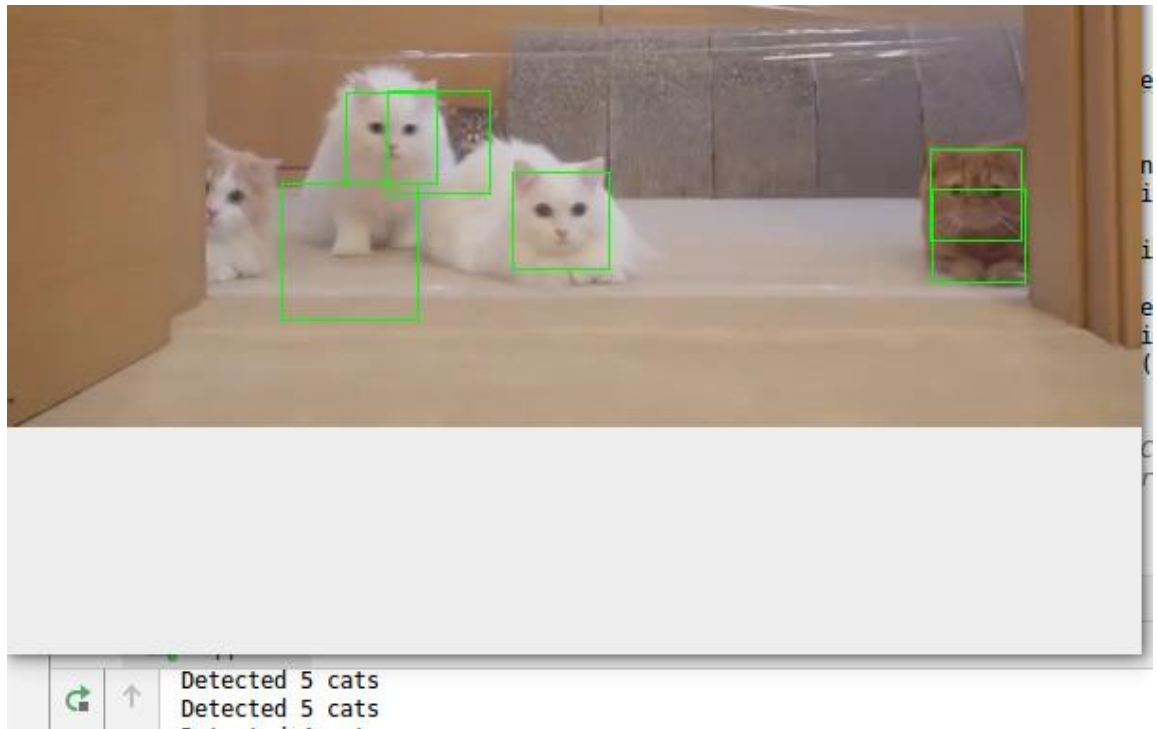


Рисунок 3.2 – Приклад роботи бібліотеки OpenCV на тваринах

Розглянемо також і інший приклад.

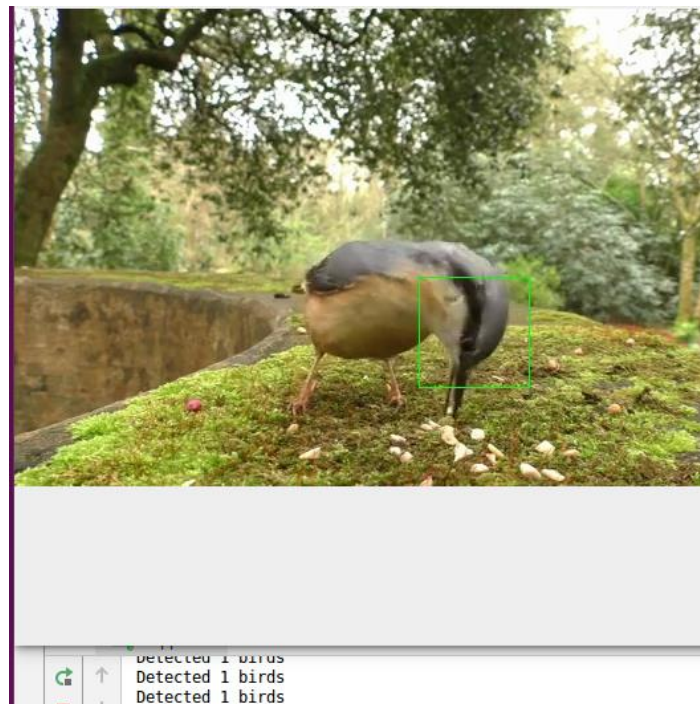


Рисунок 3.3 – Приклад роботи бібліотеки OpenCV на птахів

Підрахунок тварин та птахів може бути використаний для їх відстежування та догляду в притулках та заповідниках.

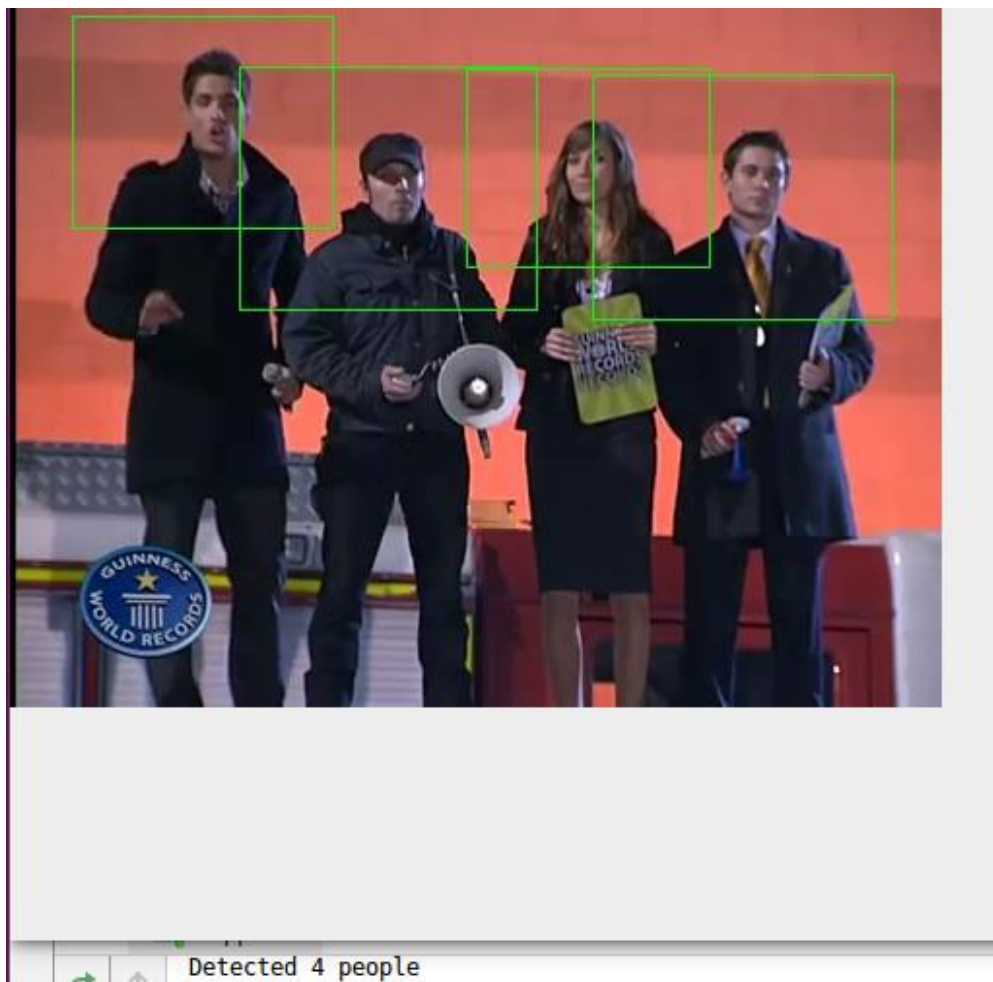


Рисунок 3.4 – Приклад роботи бібліотеки OpenCV на людях на мітингу

В наведеному прикладі реалізовано визначення верхньої частини людського тіла. Також можливе виокремлення окремих частин тіла людини, такі як очі, руки, вуха, лише обличчя або у повний зріст.



Рисунок 3.5 – Приклад роботи бібліотеки OpenCV на людях в магазині

На рисунках 3.4 та 3.5 демонструється можливість виокремлення людини як об'єкта на зображенні для відрахунку людей у публічних місцях, таких як магазин, або під час мітингу.

3.2Схема розв'язання на основі Tensorflow

Фреймворк Tensorflow є дуже популярним та доступним серед розробників. Він базується на використанні нейронних мереж. В даному випадку використовувалися регіонально-згорткові нейронні мережі. [17]

Розглянемо схему розв'язання задачі з використанням фреймворку Tensorflow та бібліотеки OpenCV на рисунку 3.6.

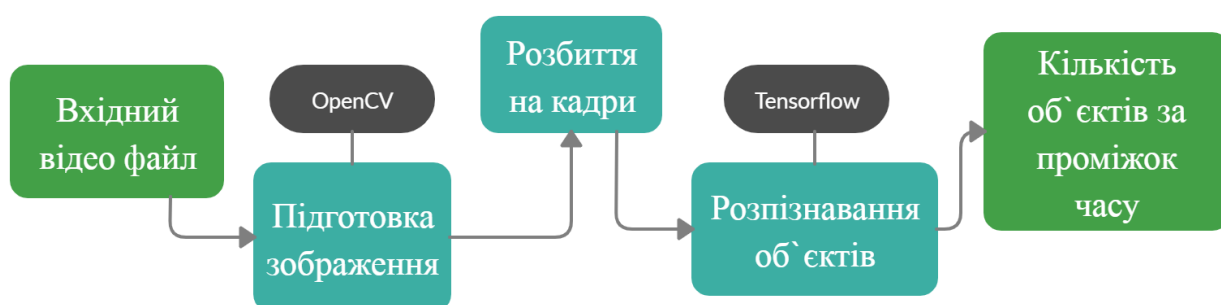


Рисунок 3.6 – Схема розв'язання задачі з використанням фреймворку TensorFlow

Спочатку отримується вхідний файл відео. Після цього йде підготовка хображення за допомогою бібліотеки OpenCV та розбиття на кадри.

За допомогою фреймворку Tensorflow[13] та нейронних мереж відбувається розпізнавання об'єктів. Для тренування нейронної мережі був взятий готовий набір даних.

Розглянемо етапи рішення задачі з використанням Tensorflow на рисунку 3.7

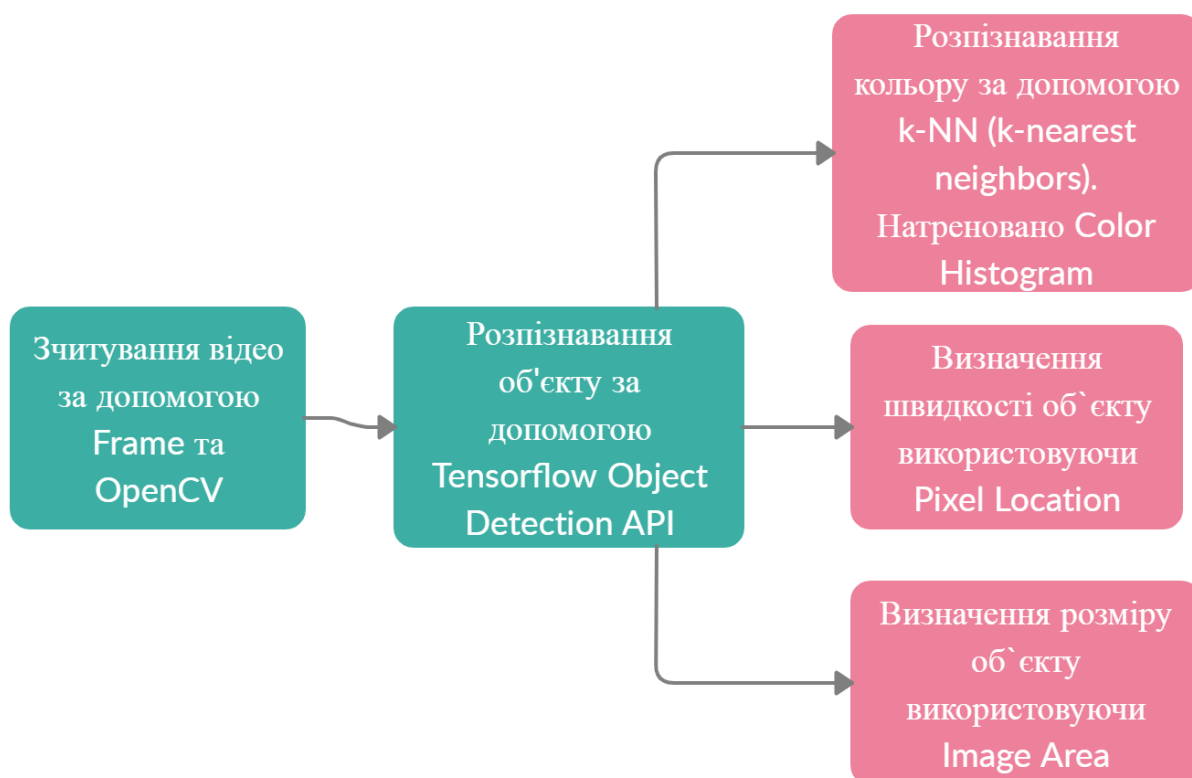


Рисунок 3.7 – Етапи рішення задачі з використанням Tensorflow

Було визначено колір машини, її розмір та приблизну швидкість пересування.

Також необхідно зазначити, що для розпізнавання об'єктів була використана регіонально згорнута нейронна мережа.[15]

Вона дає досить точні результати, хоча має деякі недоліки пов'язані зі швидкістю впровадження.

Висновки до розділу 3

У цьому розділі було висвітлено схеми етапів розв'язання задачі з виокремлення об'єкту на зображенні для того, щоб полегшити розробнику розробити подібний функціонал у більш короткий час та прикладаючи менше зусиль.

4 ОБЧИСЛЮВАЛЬНІ ЕКСПЕРИМЕНТИ

Для того, щоб відобразити результати роботи проєкту було взято дві тестові задачі. Перша задача – це розпізнавання та підрахунок машин на відео з дороги, та друга - відео з підземного паркінгу.

У роботі порівнюються результати обох програмних забезпечень, що були застосовані на однакових тестових відеокадрах.

4.1 Тестова задача

В рамках тестової задачі були взяті два відео у вільному доступі для відображення роботи програм на різних ділянках доріг, щоб визначити кількість автомобілів, які проїхали за проміжок часу.[11]

Для першого прикладу було взяте відео дороги з камери спостереження. Для другого прикладу було взяте відео з підземного паркінгу.[3] Програмні забезпечення, що відрізняються способами, які вони використовують для визначення об'єктів на відеокадрах, були застосовані на однакових відео для більш точного та хручного порівняння результатів їх роботи.

Перше програмне забезпечення працює з використання нейронних мереж та фреймворку Tensorflow[10]. Друге програмне рішення працює з використанням каскадних методів, що втілені у бібліотеці OpenCV.

4.2 Результат роботи та порівняння застосованих підходів

Результат роботи програмного забезпечення з використанням нейронних мереж на тестовому завданні номер один, що представляє собою дорогу зняту на камеру спостереження, як показано на рисунку 4.1.



Рисунок 4.1 – Тестова задача 1 з використанням Tensorflow

Результат роботи програмного забезпечення з використанням каскадних методів, що втілені у бібліотеці OpenCV, що представляє собою визначення машин на дорозі, знятої на камеру спостереження, як показано на рисунку 4.2.

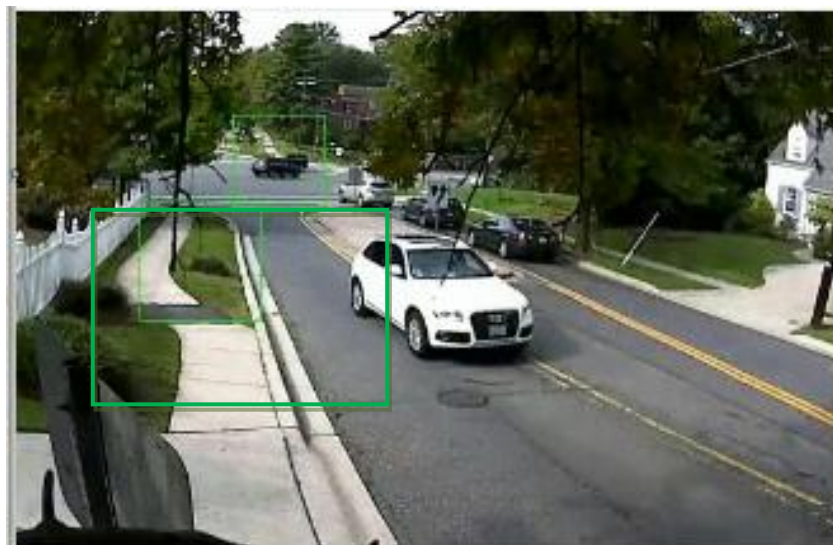


Рисунок 4.2 – Тестова задача 1 з використанням OpenCV

Результат роботи програмного забезпечення з використанням нейронних мереж на тестовому завданні номер один, що представляє собою підземний паркінг, як показано на рисунку 4.3.



Рисунок 4.3 – Тестова задача 2 з використанням Tensorflow

Результат роботи програмного забезпечення з використанням каскадних методів, що втілені у бібліотеці OpenCV[3], що представляє собою визначення машин на підземному паркінгу, як показано на рисунку 4.4.

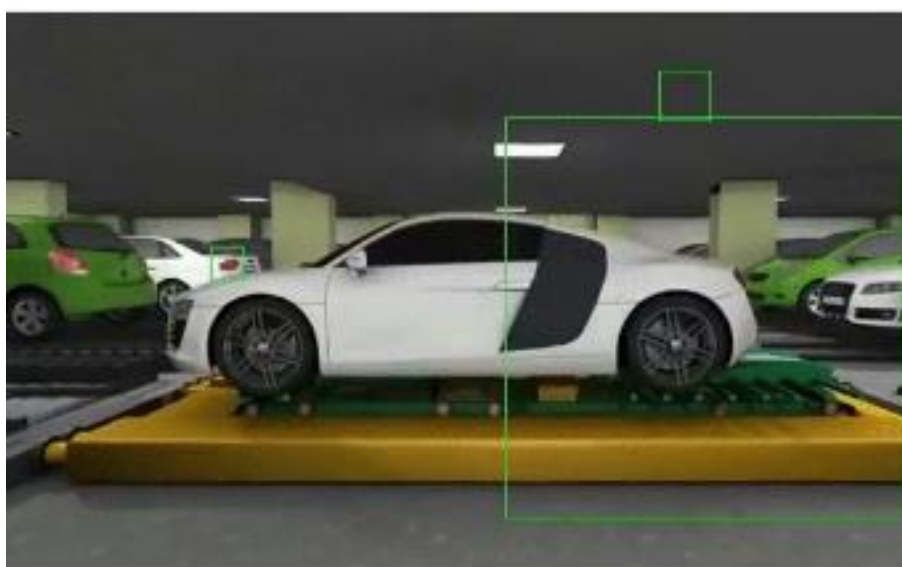


Рисунок 4.4 – Тестова задача 1 з використанням OpenCV

Розглянемо порівняння роботи програмних забезпечень, розроблених різними методами на таблиці, що відображає задачу, кількість машин, що була зафіксована за проміжок часу програмним забезпеченням з використанням бібліотеки OpenCV та фреймворку Tensorflow у порівнянні із точною кількістю машин на відео, як показано на таблиці 4.1.

Таблиця 4.1 – Точність виконання додатків

№	Задача	к-сть машин за проміжок часу (OpenCV)	к-сть машин за проміжок часу (Tensorflow)	Точна кількість
1.	Дорога	20	25	26
2.	Паркінг	11	10	10

Оцінивши вищевказану таблицю, можна зробити висновки, що технологія з використанням нейронних мереж та фреймворком Tensorflow працює краще, ніж програмне забезпечення з використанням бібліотеки OpenCV, хочі можна відмітити, що обидва способи показали гідні результати.[2]

Висновки до розділу 4

У цьому розділі були розкриті результати роботи програмних забезпечень, що були створені різними методами, розглянутими вище. Була проаналізована їх робота на прикладі тестових завдань, якими є визначення кількості машин на дорозі на закритому паркінгу.

Обидва методи показали хороші результати, однак можна зазначити, що програмне забезпечення, розроблений із використанням нейронних мереж, показав дещо кращі результати.

5 РОЗРОБКА СТАРТАП ПРОЄКТУ

Розроблене рішення у вигляді схеми використання розробленого програмного забезпечення для всіх етапів розв'язання задачі по виокремленню об'єкту на зображенні є надзвичайно затребуване в наш час.

Визначення об'єкту на різних відеокадрах та статичних зображеннях використовується майже усюди в нашому повсягденному житті. Це може бути розпізнавання вашого обличчя при розблокуванні смартфона, чи на вході у офіс чи лабораторію. Це також необхідно для відстежування тварин та птахів у заповідниках, людей у будь-яких публічних місцях для того, щоб уникати небажаних осіб. Окрім цього, такі рішення застосовуються для виокремлення таких об'єктів як машини, щоб відстежувати навантаженість доріг, затори, аварії, контролювати підвищення швидкості пересування транспорту тощо.

Загалом, можливо розпізнавати будь-який об'єкт, що забажає людина, для цього потрібно лише правильно налаштувати свою систему, «навчити» її виокремлювати на зображенні певні об'єкти.

Для того, щоб довести важливість створення чіткої схеми розробки програмних продуктів для розпізнавання об'єктів на зображенні, необхідно проаналізувати всі позитивні та негативні сторони, оцінити ризики та слабкі і сильні сторони, та визначити цільових покупців та конкурентів даного стартапу.

5.1 Опис ідеї проєкту

Для кожного успішного та вдалого проєкту необхідна чудова ідея. Отже, для розробки даного стартапу була використана ідея, що описана у таблиці 5.1.

Таблиця 5.1 — Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розроблене рішення у вигляді схеми використання розробленого програмного забезпечення для всіх етапів розв'язання задачі по виокремленню об'єкту на зображенні.	Визначення етапів розв'язання задачі по виокремленню об'єктів.	Спрощення розуміння етапів побудови додатку для розпізнавання об'єктів на відеокадрах.
	Визначення кращих засобів для визначення об'єкту.	Дозволяє зекономити час завдяки використанню простого і зрозумілого інтерфейсу
	ІТ компанії, програмісти, науковці.	Готова покрокова схема розробки додатку з основною функцією розпізнавання об'єкту на відео.

З таблиці наведеної вище можна зробити висновки, що основними користувачами обрано розробників зі сфери інформаційних технологій, які небагато знають про обробку зображень, або бажають заощадити свій час відведений для розробки бажаного програмного забезпечення із виокремлення об'єкту на відеокадрах.

Розроблена схема є вигідною для користувачів, тому що в них не буде необхідності проводити власні дослідження та аналіз щодо кращого методу визначення об'єкту на зображенні, шукати найбільш оптимальні рішення та заглиблюватись у деталі роботи кожного підходу самостійно.

Реалізовані тестові завдання на основі двох розглянутих методів показують затребуваність розв'язання таких задач. За допомогою використання цих тестових завдань, як прикладів, користувач має змогу швидко перепрограмувати програмне забезпечення для визначення цікавих йому об'єктів змінивши всього декілька параметрів у коді, таких як датасет у методі з використанням нейронних мереж, на

основі яких вони «навчаються», або .xmlкласифікатора. Готові набори даних та .xmlкласифікатора можна знайти у вільному доступі або створити свої власні.

Розробка програмного продукту на основі існуючої схеми значно пришвидшить процес створення бажаного користувачем проєкту та заощадить час.

Після опису ідеї для стартап-проєкту будуть розглянуті та визначені характеристики продукту.

Таблиця 5.2 — Визначення сильних, слабких та нейтральних характеристик ідеї проєкту

№ п/п	Техніко- економічні характери- стики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Розроблюваний проєкт	Конкурент 1	Конкурент 2	Конкурент 3			
1	Точність	середня	висока	низька	середня		+	
2	Собівартість	низька	середня	висока	низька			+
3	Використання додаткового обладнання	Ні	Ні	Так	Так		+	
4	Кросплатформність	Ні	Так	Ні	Ні	+		

У таблиці 5.2 вище порівнюється розроблений проєкт з деякими конкурентами. Конкурент 1 — Cogniac, 2 — Usu, 3 — ObjectDetection 2.0.

Проаналізувавши цю таблицю можна зрозуміти що сильною стороною розробленого проєкту є невисока собівартість. Нейтральними сторонами є точність

визначення об'єкту на відеокадрах та використання додаткового обладнання. Останнє не є абсолютно негативною стороною, тому що із додаткового обладнання необхідні лише камери, з яких потім потрібно брати відео для аналізу. Однак, це залишається на розсуд користувача, так як програма може працювати на будь-якому готовому відео, узятим, наприклад, із мережі.

Беручи до уваги вищевказані фактори, розроблений програмний продукт можна вважати конкурентоспроможним.

5.2 Технологічний аудит ідеї проекту

В даному пункті необхідно провести аудит технологій проекту, які повинні допомогти реалізувати ідею (тобто аудит технологій застосованих для розробки програмного забезпечення).

По-перше, необхідно визначити технології, що можуть бути використані для розробки (таблиця 5.3).

Таблиця 5.3 — Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології реалізації	Наявність техно логій	Доступність техно логій
1	Бібліотека обробки зображень	OpenCV	Наявна	Доступна
		Yolo	Відсутня	Доступна
2	Засоби розробки нейронних мереж	TensorFlow	Наявна	Доступна
		OpenNN	Відсутня	Доступна

Таблиця 5.3(продовження)

3	Засоби візуалізації статистичних даних	Chart	Наявна	Доступна
		Flot	Наявна	Доступна
		ShieldUI	Наявна	Доступна
Обрані технології реалізації ідеї проекту: OpenCV, Tensorflow, Chart, Flot, ShieldUI				

Проаналізувавши наведену таблицю, можна зробити висновок, що для розробки бажаного програмного забезпечення існують аналоги, тож є вибір у тому, чкі технології використовувати для реалізації. За бажанням, можна використовувати і вищенаведені технології, що не були використані для написання даного програмного забезпечення.

Для обробки зображень та виокремлення об'єкту на відеокадрах першим способом було використано бібліотеку OpenCV. Вказана бібліотека є одною із найбільш популярних та доступних в наш час для рішення подібних задач. При зміні декількох параметрів можна визначати різні об'єкти, які потрібні розробнику.

Для виокремлення об'єкту на відеокадрах другим способом було використано фреймворк TensorFlow та регіонально-згорнутої нейромережі (R-CNN). TensorFlow має підтримку від компанії Google та велику популярність серед розробників. Існує багато видів нейронних мереж, однак R-CNN було обрано через те, що вона дає більшу точність, хоча і важча у впровадженні.

Були розроблені два схематичних рішення розв'язання поставленої задачі за двома підходами на основі класифікаторів та нейронних мереж.

Для візуалізації роботи обох методів були обрані тестові завдання у вигляді визначення машин на відеокадрах, що були взяті з мережі та були у вільному доступі. Прикладами слугували відео дороги та паркінгу.

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення можливостей ринку, що можливо використати для ринкового впровадження проекту та ринкових загроз, що можуть перешкоджувати реалізації проекту, дозволяє спланувати напрями розвитку проекту з урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Проведемо аналіз попиту (таблиця 5.4).

Таблиця 5.4 — Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	Біля 4
2	Загальний обсяг продаж, грн/ум.од	12 000 грн за ум.од.
3	Динаміка ринку (якісна оцінка)	Очікується зростання
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	15

Проаналізувавши вищевказану таблицю видно, що інвестування коштів у пропонований проект є вигідним. Окрім цього, необхідно відмітити, що відсутні

Аналізуючи попередню таблицю, зрозуміло, що вкладання коштів в цей проект є вигідним. Слід звернути увагу, що обмежень для входу і до стандартизації немає, що полегшує ситуацію. До того ж, на ринку очікується зростання в цій сфері.

Визначимо цільову аудиторію програмного продукту, її поведінки та вимог (таблиця 5.5).

Таблиця 5.5 — Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінність у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Можливість скористатися схемою етапів для побудови власного	Розробники	Не передбачається	Проста та доступна схема побудови ПЗ для визначення об'єктів на відеокадрах
2	швидкого і якісного ПЗ із визначення об'єктів на зображенні	ІТ компанії	Не передбачається	Проста та доступна схема побудови ПЗ для визначення об'єктів на відеокадрах

Проаналізувавши останню таблицю, цільовою аудиторією є розробники та ІТ компанії. Окрім цього, не очікується відмінностей у поведінці у цих аудиторій, так як вони частково перетинаються.

Проаналізуємо ринкове середовище. Визначимо фактори загроз (таблиця 5.6) та можливостей (таблиця 5.7), які можуть перешкоджати або загрожувати впровадженню проєкту.

Таблиця 5.6 — Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Ризик технології	Використання R-CNN може сповільнити впровадження програмного продукту у порівнянні із іншими нейронними мережами.	Пришвидшення впровадження завдяки готовій схемі етапів розробки програмного забезпечення.
2	Недоступність платформи	Відсутність можливості запускати програмне забезпечення на мобільних пристроях, планшетах тощо. Відсутня кросплатформенність.	Розробка ПЗ для інших платформ (наприклад, iOS, Android).

У вищевизначеній таблиці було висвітлено основні загрози впровадження на ринок даного програмного продукту. Вважається, що найбільші ризики може принести відсутність кросплатформенності, тоді як у конкурентів може бути реалізація на інші пристрої.

Окрім цього, використання R-CNN може сповільнити впровадження програмного продукту у порівнянні із іншими нейронними мережами. Тому потрібно розглядати варіант використання інших нейронних мереж, надаючи перевагу швидкості впровадження точності виокремлення об'єктів на відеокадрах.

Розглянемо фактори можливостей у таблиці 5.7.

Таблиця 5.7 — Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії

Таблиця 5.7 (продовження)

1	Розвиток технологій та підвищення точності виокремлення об'єктів на відеокадрах, використання розробленої схеми етапів для навчання у школах та університетах.	Є можливість, що при збільшенні зацікавлених компаній та приватних розробників, збільшить і попит на розроблений проєкт.	Посилення рекламної кампанії, а саме розповсюдження її у соціальних мережах.
---	--	--	--

Одним із найбільш важливих факторів є можливість розповсюдження проєкту в інститутах. Така реклама забезпечить збільшення кількості зацікавлених розробників.

Розглянемо типи конкуренції у таблиці 5.8.

Таблиця 5.8 — Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
Тип конкуренції	Олігополія	Розробка більшої кількості схем етапів для інших методів виокремлення об'єктів на відеокадрах.

Таблиця 5.8 (продовження)

За рівнем конку rentної боротьби	Глобальна	
За галузевою ознакою	Міжгалузева	
Конку rentція за видами товарів	Товарно-видова	
За характером конку rentних переваг	Нецінова	
За інтенсивністю	Не марочна	

Для того, що провести більш розширений аналіз конкуренції, скористаємося аналізом конкуренції в галузі за М. Портером.

Наведений аналіз можливо розглянути у таблиці 5.9.

Такий аналіз конкуренції є непоганим, але неповним. Саме тому варто провести аналіз конкуренції в галузі за М. Портером (таблиця 4.9).

Таблиця 5.9 — Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти	Потенційні конкуренти	Постачальник	Клієнти	Товари-замінники
	Навести перелік прямих конкурентів	Визначити бар'єри входу в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загрози з боку замінників

Таблиця 5.9 (продовження)

Висновки:	Найближчих трьох основних конкурентів. Найсерйознішими із них є CogniastObjectDetection 2.0.	Бар'єрами є використання R-CNN, тому що вони сповільнюють впровадження проєкту.	Постачальник відсутній.	Зручність у використанні, Досить висока точність завдяки R-CNN.	Загроза більш швидкого впровадження конкурентами або розробка більш дешевого аналогу.
-----------	--	---	-------------------------	---	---

Проаналізувавши останню таблицю, ми можемо зробити висновки, що проєкт має двох основних конкурентів.

Враховуючи досить високу точність роботи і зручність розробленої схеми етапів для програмного забезпечення, можна вважати, що проєкт має можливість поширитися на ринку.

Беручи за основу результати цього аналізу можна виокремити основні фактори конкурентоспроможності продукту. Розглянемо їх у вигляді таблиці 5.10.

Таблиця 5.10 — Обґрунтування факторів конкурентоспроможності

№ п/п	Фактори конкурентоспроможності	Обґрунтування
-------	--------------------------------	---------------

Таблиця 5.10(продовження)

1	Функціонал	Наявність визначення об'єктів на відеокадрах на прикладі машин. Визначення швидкості пересування об'єктів, їх колір і розміру.
2	Вартість	Пропонується випуск продукту з нижчою ціною у порівнянні із конкурентами.
3	Не потребує додаткового обладнання	Для роботи програмного забезпечення не є обов'язковим наявність додаткового обладнання. Однак користувач має можливість використання власних камер для того, що використовувати із них відеоматеріали.

Із таблиці, що була наведена вище, можна побачити, що відсутність необхідності у додатковому обладнанні та достатньо висока точність виокремлення об'єктів надають перевагу над конкурентами.

Розглянувши такі висновки, ми можемо провести аналіз сильних та слабких сторін у порівнянні з конкурентами.

Результати можна переглянути у вигляді таблиці 5.11.

Таблиця 5.11 — Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентноспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з розроблюваним продуктом						
			-3	-2	-1	0	1	2	3

Таблиця 5.11(продовження)

1	Функціонал	16			+				
2	Вартість	18	+						
3	Не потребує додаткового обладнання	17				+			

Аналізуючи вищенаведену таблицю, можна побачити, що найбільшу кількість балів було набрано завдяки фактору «вартість». Тільки один з факторів є нейтральним.

Завершимо ринковий аналіз таблицею можливостей впровадження розробленого проєкту у вигляді SWOT аналізу, який являє собою слабкі та сильні сторони, а також можливості та загрози для проєкту у таблиці 5.12.

Заключною таблицею ринкового аналізу можливостей впровадження розробленого програмного продукту є аналіз SWOT, що представляє собою сильні та слабкі сторони, можливості і загрози для проєкту в одній таблиці (таблиця 6.12).

Таблиця 5.12 — SWOT-аналіз стартап-проєкту

Сильні сторони: потребує додаткового обладнання, точність, вартість	не	Слабкі сторони: Відсутність кросплатформеності. Неможливість використання програмного забезпечення на мобільних телефонах, планшетах тощо.
Можливості: Поширення проєкту завдяки навчальним закладам та розширеній рекламній кампанії, покращення точності та реалізація ПЗ на різних платформах.		Загрози: перевага конкурентів, що мають більшу кількість клієнтів та кросплатформеність.

Вищенаведена таблиця по суті є узагальненням вищесказаного.

У випадку, якщо клієнти будуть надавати перевагу кросплатформним програмним забезпеченням планується розширити функціонал для роботи на різних пристроях.

Так як є SWOT-аналіз, можна скласти таблицю 5.13 альтернатив ринкового впровадження стартап-проекту.

Таблиця 5.13 — Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива	Ймовірність отримання ресурсів	Строки реалізації
1.	Розробка виокремлення об'єктів у реальному часі	90%	1
2.	Розробка схеми етапів для більш нових та точних методів виокремлення об'єктів	40%	1,5

Беручи до уваги останню таблицю, можна зробити висновки, що кращим рішенням буде взяти першу альтернативу, тому що на неї буде використано менше часу та вона простіша у виконанні, однак є дуже затребуваною.

В разі успіху першої альтернативи, можливо почати розробку другої альтернативи, так як технології розвиваються надзвичайно швидко і ми повинні відповідати всім вимогам сучасного ринку.

5.4 Розробка ринкової стратегії проекту

Перше, що необхідно зробити для розробки ринкової стратегії – це визначити стратегію охоплення ринку. Розглянемо її у таблиці 5.14.

Таблиця 5.14 — Вибір цільових груп потенційних споживачів

№ п/п	Опис цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Приватні особи (розробники)	+	Середній	Висока	Складно
2	ІТ компанії	+	Високий	Вище середнього	Складно
3	Навчальні заклади	+	Вище середнього	Практично немає	Просто
Які цільові групи обрано: 1, 2					

Аналізуючи таблицю, наведену вище, можна зазначити, що основними цільовими групами потенційних клієнтів є ІТ компанії та приватні особи - розробники.

Аудиторія у вигляді навчальних закладів також підходить дуже добре, але не є основною, зважаючи на фінансовий стан людей, які тільки навчаються.

Отже, беручи до основи висновки, що були зроблені вище, необхідно відобразити базову стратегію розвитку. Розглянемо її у таблиці 5.15.

Таблиця 6.15 — Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охопле ння ринку	Ключові конкурентоспро можні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Розробка схем етапів виключно для різних типів нейромереж	Стратегія дифер енційованого маркетингу	Функціонал, Вартість, Не потребує додаткового обла днання	диференціація

Зважаючи на те, що в планах є співпраця з декількома сегментами ринку, було обрано стратегію диференційованого маркетингу.

Так як проект пропонує не найбільш поширений підхід, а тобто схему етапів рішення задачі виокремлення об'єкту на відеокадрах, було обрано диференціацію, як базову стратегію.

Визначення базову стратегію конкурентної поведінки, що наведена у вигляді таблиці 5.16.

Таблиця 5.16 — Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект "першопрохідцем" на ринку?	Чи буде компанія Шукатинових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента	Стратегія конкурентної поведінки
1	Ні	Шукатинових споживачів та забирати існуючих	Так	Стратегія наслідування лідери

Безсумнівно, запропонований розроблений проєкт не є першопрохідцем на ринку. Існує багато програмних рішень для визначення об'єктів на відеокадрах та статичних зображеннях. Однак, перевагою є добре відпрацьована та розроблена схема етапів, які повинна допомагати розробникам швидко і без зайвих зусиль реалізовувати властні програмні продукти, навіть не сильно заглиблюючись у методи сегментації та обробки зображення у цілому.

Найкращим рішенням стосовно користувачі є як і знаходження нової аудиторії, так і збір існуючої аудиторії конкурентів.

Беручи до уваги вимоги користувачів, конкурентну поведінку та стратегію розвитку, розробимо стратегію позиціонування, щоб допомогти користувачам ідентифікувати проєкт.

Розглянемо це на таблиці 5.17.

Таблиця 5.17 — Визначення стратегії позиціонування

№ п/п	Вимоги до товару цілової аудиторії	Базова стратегія розвитку	Ключові конкурентні переваги проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Низька ціна, точність, простота у використанні	Диференціація	Висока точність завдяки регіонально-згорнутої нейромережі (R-CNN), порівнянню низької ціни, розроблена схема етапів розробки подібних програмних продуктів з використанням різних засобів.	Простота, ясність, зручність, функціональність, точність.

Вищенаведена таблиця яскраво відображає проєкт у цілому, та характеризує його як зрозуміле, точне, зручне та недороге рішення, беручи до уваги конкурентів та вимоги користувачів.

5.5 Розроблення маркетингової програми

Розглянемо маркетингову програму. Першим кроком сформулюємо маркетингову концепцію товару. Продемонструємо це на таблиці 5.18.

Таблиця 5.18 — Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Функціонал	Кросплатформне програмне забезпечення та схеми етапів розв'язання задачі під кожний пристрій	Існує у невеликої кількості конкурентів
2	Вартість	Невисока ціна для клієнтів	Нижча ціна у порівнянні з конкурентами
3	Незалежність системи від додаткових пристроїв	Не потребує додаткового обладнання	Необхідний лише браузер та комп'ютер

Розробимо трирівневу маркетингову модель товару. Розглянемо її на таблиці 5.19.

Таблиця 5.19 — Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Схема етапів рішення задачі з виокремлення об'єктів на відеокадрах.

Таблиця 5.19(продовження)

II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Не потребує додаткового обладнання	Нм	Тх
	Низька ціна		
	Простий у розумінні і використанні		
	Якість: продукт протестовано вручну		
	Маркування: Немає		
Компанія: назва “ObjDet”			
III. Товар із підкріпленням	Придбання товару завдяки оформленню підписки, регулярна підтримка додатку		
За рахунок чого потенційний товар буде захищено від копіювання: обфускація коду			

У вищевказаній таблиці було висвітлено три рівні моделі проєкту.

В попередній таблиці було наведено три рівні моделі товару. Це Схема етапів рішення задачі з виокремлення об'єктів на відеокадрах.

Також необхідно зазначити, що товар можна отримати завдяки підписці, а це означає регулярне підтримання програмного забезпечення. Товар буде захищено допомогою обфускації коду від копіювання та неправомірного розповсюдження.

Визначимо цінові межі та розглянемо їх на таблиці 5.20.

Таблиця 5.20 — Визначення меж встановлення ціни

№ п/п	Рівень цін на товари - замінн ики	Рівень цін на товари - аналог и	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар
1.	14000	20000	30000	4000-12000

Визначимо систему збуту, яка є оптимальною для даного проекту та розглянемо її на таблиці 5.21.

Таблиця 5.21 — Формування системи збуту

№ п/п	Специфіка закупівельної поведінки клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Покупка та щомісячна або щорічна підписка	Продаж	0 (напряму)	Власна

Вищевказана таблиця відображає систему збуту. В ній описано, що користувачі будуть мати змогу купувати щомісячну або щорічну підписку та що система збуту буде представлена без посередників.

Розглянемо фінальний етап формування маркетингової програми, а саме розробку концепцій маркетингових комунікацій на таблиці 5.22.

Таблиця 5.22 — Концепція маркетингових комунікацій

№	Специфіка	Канали	Ключові позиції,	Завдання
---	-----------	--------	------------------	----------

п/п	ведінки цільових клієнтів	комунікацій, якими користуються цільові клієнти	обрані для позиціонування	рекламного повідомлення
1.	Доступ по посиланню	Інтернет	Простота, точність, низька ціна, функціонал, відсутність додаткового обладнання	Висвітлити призначення проекту просто і ясно, довести зручність користування розробленими схемами етапів та розробленим програмним забезпеченням.

З вищевказаної таблиці можна зробити висновки, що користувач буде мати змогу перейти по посиланню на web-додаток, створити свій профіль та оформити підписку, яку він бажає: щомісячну або щорічну.

Висновки до розділу 5

У цьому розділі були висвітлені головні аспекти виходу на ринок розробленого рішення, який є web-додатком та включає в себе приклади вирішення задач з виокремлення об'єктів на відеокadraх та схему етапів по розв'язанню поставленої задачі за допомогою різних методів, а саме завдяки бібліотеці OpenCV та фреймворку Tensorflow.

Окрім цього було детально розписано ідею проекту та зроблено аналіз ринкових можливостей та технологічний аудит запуску стартап-проекту.

Наступним кроком було висвітлено ринкову стратегію проекту. Також, для стартапу було розроблено маркетингову програму.

В якості альтернативи було обрано розробку схеми етапів та шаблон програмного забезпечення виключно з використанням різних типів нейромереж. Також було запропоновано втілити функціонал визначення шуканих об'єктів у реальному часі.

Наступним кроком було визначено слабкі та сильні сторони прокту. Серед сильних сторін є використання регіонально-згорнутої нейромережі, однак це є слабким місцем, адже уповільнює впровадження проєкту.

Далі було проаналізовано ризики і визначено, що необхідно робити при їх з'явленні, включаючи розширення проєкта для більшої кількості пристроїв. Беручи це до уваги, було виконано SWOT-аналіз.

Під час проведення аналізу проєкту було виявлено аналогічні проєкти конкурентів, серед яких найбільш серйозними є Cogniasta ObjectDetection 2.0.

Порівняння цих програмних продуктів із розробленим показало, що в усіх розглянутих проєктів є переваги і недоліки, та розроблений проєкт є конкурентоспроможним.

Беручи до уваги потенційну аудиторію клієнтів та невисокі бар'єри входження, можна зробити висновок, що розроблений проєкт має гарні перспективи.

ВИСНОВКИ

Досліджено підходи до розв'язання задачі виокремлення об'єкту на відеокадрах. Визначено доцільність використання навчених класифікаторів та регіонально-згорнутої нейромережі (R-CNN). Були висвітлені позитивні та негативні сторони використання саме цього типу нейронних мереж.

Вивчено функціонал бібліотеки OpenCV. Відібрано модулі, які відповідають всім етапам поставленої задачі. Розглянуто, як працює вищевказана бібліотека та висвітлені методи, що вона використовує для визначення об'єктів на зображеннях. Також було розглянуто методи, які можна використовувати для покращення якості зображення для отримання кращих результатів її роботи.

Визначено основні етапи при роботі з Tensorflow. Досліджено нейронну мережу, яку доцільно застосовувати для виокремлення об'єкту на відеокадрах.

Розглянуто основні принципи роботи нейронних мереж, їх типи, висвітлені позитивні та негативні сторони.

Розроблено 2 схематичних рішення розв'язання поставленої задачі за двома підходами на основі класифікаторів та нейронних мереж.

Проведено обчислювальні експерименти для порівняльного аналізу результатів розпізнавання.

Було описано ідею проєкту та сформовано технологічний аудит ідеї проєкту, проаналізовано ринкові можливості запуску стартап-проєкту, розроблені ринкові стратегії проєкту, розроблені маркетингові програми стартап-проєкту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Lukanitsa A. A. Digital processing / A.A. Lukanitisa, A.G. Shishkin - Moscow: ISSC Press, 2009. – 518с.
2. Yilmez A. Object tracking: A survey /A.Yilmez, O. Javied, M. Shah. - ACM Computing Surveys, 2007, vol. 34, no. –95с.
3. Martínaz-Marten E. Robast motion detection in real-life scenarios./ E. Martínaz-Marten, A. Poboil– Springer-Verliag London, 2016. – 255с.
4. Виокремлення об'єктів. [Електронний ресурс]/Режим доступу:<https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/>
5. Novoseltsev IV Hybrid Neural Networks / IV Novoseltsev., N.G. Aksaik -M: Information Processing Systems, 2005. - Vip. 4. –70 с.
6. Комп'ютерний зір. [Електронний ресурс]/Режим доступу:<https://www.amazon.com/Computer-Vision-Modern-Approach-2nd/dp/013-608592X>
7. Prat W. Digital image processing / W. Prat - М .: Mir, 1985. - 490 с.
8. Нейронні мережі. [Електронний ресурс]/Режим доступу:<https://www.techradar.com/news/what-is-a-neural-network>
9. Soifier V.A. Methods of computer image processing / V.A. Soifier - Moscow: FIZMAOTLIT, 2005.— 790 с.
10. Теоретичні основи цифрової обробки. [Електронний ресурс]/Режим доступу: <https://www.twirphx.com/file/146812600/>
11. Gonzaletz R. Digital image processing / R. Gonzaletz, R. Wood - М .: Technosphere, 2015. — 1125 с.
12. Anisimov B. V. Recognition and digital image processing // BV Anisimov, VD Kurgenov, VK Zobin - М .: Vysh., 1953. — 232 с.

13. Shepir L. Computer vision // L. Shepir, D. Stockman - M.: Binomm. Laboratory of knowledge, 2005. — 767 c.
14. Y. V. Vizeilter. Processing of images in computer vision. // Y. V. Vizeilter, S. Y. Zhetov, - Moscow: Fizmatkniga, 2009. — 698 c.
15. Zheravlov Y.I. Image recognition and image recognition // Y. I. Zheravlov, I.B. Guravich - Moscow: Science, 1990. — 81 c.
16. JaneD. Digital image processing / D. Yan - M.: Technosphere, 2005. — 598 c.
17. OpenheimO. Digital processing/O. Openheim, V. Safir - M.: Technosphere, 2008. - 1658 c.
18. Forsuh R. Computer Vision: A Modern Approach (2nd Edition) / R. Forsuh, J. Polce - Person, 2011. - 982 p.

ДОДАТОКА

ВИОКРЕМЛЕННЯ ОБ'ЄКТІВ НА ВІДЕОКАДРАХ

Тези на конференцію

«Сучасні проблеми наукового забезпечення енергетики»

УКР.НТУУ"КПІ ім. Ігоря Сікорського" _ТЕФ_АПЕПС_42103_18Б 13-1

Аркушів 1

Київ 2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ

Матеріали XVII Міжнародної
науково-практичної конференції
МОЛОДИХ ВЧЕНИХ ТА СТУДЕНТІВ
м. Київ, 23-26 квітня 2019 року,

ТОМ 2



Київ- 2019

МОСКАЛЕНКО Ю.В., аспірант	
Машинне навчання для розв'язання логічних головоломок.	93
БАРАНИЧЕНКО О.М., магістрант гр. ТВ-71мн	
Керівник - доц., к.т.н. Шаповалова С.І.	
Веб-середовище для моделювання процесів міжагентної взаємодії в мережах Smart Grid.	94
ШВАЙКА Д.А., магістрант гр. ТР-81мн	
Керівник - доц., к.ф.-м.н. Тарнавський Ю.А.	
Використання техніки Structure from Motion в системі навігації.	95
ХАРАБАР В.В., магістрант гр. ТВ-81мн	
Керівник - доц., к.т.н. Гагарін О.О.	
Система оцінювання екологічних збитків у мережі АЗС.	96
ОЛЕКСІЙ А.О., магістрант гр. ТВ-82	
Керівник - доц., к.т.н. Гагарін О.О.	
Інтелектуальна система розпізнавання та передбачення намірів користувача.	97
МЕЛЬНИЧЕНКО А.В., магістрант гр. ТВ-81мн	
Керівник - ст.викл., к.т.н. Шалденко О.В.	
Проблема формування схеми замкнутого простору у системах внутрішньої навігації.	98
МАРУНЯ А.В., магістрант гр. ТВ-81мн	
Керівник - доц., к.т.н. Гагарін О.О.	
Застосування нейронних мереж в мобільних застосунках.	99
МАРИЧ Т.І., магістрант гр. ТВ-81мн	
Керівник - доц., к.т.н. Шаповалова С.І.	
Генерація елементів цифрового контенту на основі аналізу тексту.	100
КРЮЧКОВСЬКА А.В., магістрант гр. ТВ-81мн	
Керівник - ст.викл., к.т.н. Шалденко О.В.	
Програмний інструментарій виокремлення заданих об'єктів на зображенні.	101
КРУГЛИК Д.С., магістрант гр. ТВ-81мн	
Керівник - доц., к.т.н. Шаповалова С.І.	
Система розпізнавання жестів рук для людино-машинної взаємодії.	102
КОНКІНА Н.С., магістрант гр. ТВ-81мн	
Керівник - ст.викл., к.т.н. Шалденко О.В.	
Проблема вибору раціонального методу позиціонування користувача для системи навігації.	103
ЗАРИЦЬКИЙ В.П., магістрант гр. ТВ-81мн	
Керівник - доц., к.т.н. Гагарін О.О.	
Нейромережеве архітектурне рішення для обробки аудіосигналів.	104
ВИТВИЦЬКИЙ Д.А., магістрант гр. ТВ-81мн	
Керівник - ст.викл., к.т.н. Мажара О.О.	
Побудова сучасного веб-серверу на основі безсерверних технологій.	105
БРУНЬКО П.В., магістрант гр. ТВ-81мн	
Керівник - доц., к.т.н. Шаповалова С.І.	
Сегментація бур'янів на зображеннях з відеокамери наземного робота.	106
СОФІЄНКО А.Ю., студент гр. ТР-52	
Керівник - доц., к.т.н. Шаповалова С.І.	
Серверна частина системи функціонування реєстру інформаційних ресурсів.	107
СОЛОМКІН Д.Г., студент гр. ЗПІ-ЗП-63	
Керівник - ст.викл. Гайдаржи В.І.	

ПРОГРАМНИЙ ІНСТРУМЕНТАРІЙ ВИОКРЕМЛЕННЯ ЗАДАНИХ ОБ'ЄКТІВ НА ЗОБРАЖЕННІ

Виокремлення заданих об'єктів на зображенні є затребуваною прикладною задачею. Вдалим прикладом є визначення номерних знаків автомобіля на миттєвих знімках. Для розв'язання таких задач доцільно використовувати існуючий програмний інструмент, тому систематизація наявних засобів є актуальною задачею і має практичне призначення. Одним із найзатребуваніших засобів є бібліотека OpenCV[1]. Це спеціальна бібліотека - набір типів даних, функцій та класів для обробки зображень алгоритмами комп'ютерного зору. Основні модулі бібліотеки:

- Sxcore - ядро, що містить у собі базові структури даних та алгоритми, серед яких базові операції над числовими масивами, матрична алгебра, генератори випадкових чисел та базові функції 2D графіки;
- CV - модуль обробки зображень та комп'ютерного зору, а саме базові операції над зображенням, аналіз руху, відслідковування об'єктів, визначення об'єктів, калібровка камер;
- Highgui - модуль для вводу та виводу зображень та відео, створення користувацького інтерфейсу, захват зображення з камер та відео файлів, зчитування та запис статичних зображень;
- Svaux - експериментальні та застарілі функції;

За допомогою OpenCV можна робити аналіз зміщення об'єкта, 3D-реконструкцію, сегментацію зображення, визначення форми об'єкта та відслідковування його тощо. Вона вважається найкращою бібліотекою Computer Vision з відкритим вихідним кодом, доступною в даний час. Однак така бібліотека має свої недоліки. OpenCV орієнтована на мову c++, для інших мов вона може бути зовсім нереалізована, або ж реалізована лише частково. Іншим підходом є використання згорткових мереж CNN. Нейронні мережі використовуються для рішення складних задач, які потребують аналітичних обчислень подібних тим, що робить людський мозок. Найбільш поширеними застосуваннями нейронних мереж є класифікація, передбачення та розпізнавання. Існують готові пакети з реалізаціями нейронних мереж для подібних

задач, наприклад Caffle, TensorFlow, DLIB, Keras та інші. Крім цього, існує світове співтовариство спеціалістів з машинного навчання Kaggle[2]. Ресурс Kaggle містить кращі програмні рішення за допомогою яких виграно змагання з машинного навчання для задач розпізнавання зображень. В роботі досліджено концепцію та основні переваги та недоліки цих підходів та відповідного інструментарію.

Перелік посилань:

1. OpenCV library. URL: <https://opencv.org>
2. Kaggle. URL: <https://www.kaggle.com>